



Enterprise Architect User Guide

Enterprise Architect is an intuitive, flexible and powerful UML analysis and design tool for building robust and maintainable software. From requirements gathering, through analysis, modeling, implementation and testing to deployment and maintenance, Enterprise Architect is a fast, feature-rich, multi-user UML modeling tool, driving the long-term success of your software project.



Copyright © 1998-2010 Sparx Systems Pty Ltd

Enterprise Architect User Guide

Introduction

by Geoffrey Sparks

Enterprise Architect is a complete UML-based solution for analysing, designing, managing, sharing and building software systems.

Enterprise Architect User Guide

© 1998-2010 Sparx Systems Pty Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: May 2011

Publisher

Sparx Systems

Managing Editor

Geoffrey Sparks

Technical Editors

Geoffrey Sparks

Dermot O'Bryan

Simon McNeilly

Neil Capey

Sam Mancarella

John Redfern

Vimal Kumar

Howard Britten

Brad Maxwell

Steve Meagher

Special thanks to:

All the people who have contributed suggestions, examples, bug reports and assistance in the development of Enterprise Architect. The task of developing and maintaining this tool has been greatly enhanced by their contribution.

Table of Contents

Foreword	1
Part I Introduction	3
Overview	5
What You Can Do	5
Key Benefits	7
Key Features	10
Enterprise Architect Editions	13
The Trial Version	13
Editions Available	14
The Read-only 'Lite' Edition	18
Formal Statements	19
Copyright Notice	19
End User License Agreement	20
Trademarks	23
Acknowledgements	25
Order Enterprise Architect	26
Installation	27
Register a Full License	29
Help and Support	30
Available Helpfile Formats	31
Support	31
Part II Getting Started	34
Basics	35
Common Tasks	36
A Quickstart Tutorial	37
Create Project Scenario	37
Add a View To Your Model	38
Add Packages	39
Creating Diagrams	40
Creating New Elements on Diagrams	41
Adding Connectors	42
Modify Properties	43
Moving Elements, Diagrams and Connectors	44
Move Elements Within a Package.....	44
Move Elements Between Packages.....	45
Move Elements in a Diagram.....	45
Move Elements Between Diagrams.....	46
Connect a Different Element on a Diagram.....	48
Delete Elements	48
Remove Elements From a Diagram.....	49
Delete Elements From Your Model.....	49
Delete Connectors	51
Save Changes	52
Create Requirements	53
Requirement Trace Connector	55
Typical Project Roles	56
Summary of Typical Tasks	57

Business Analysts	58
Software Architects	59
Software Engineers	60
Developers	61
Project Managers	63
Testers	64
Implementation Managers	65
Technology Developers	65
Database Developers	67
User Interface Guide	69
Start Page	70
Remove Recent Projects.....	71
Learning Center	72
Main Menu	73
File	74
Edit	74
Paste Elements Submenu.....	75
View	76
View Submenus	78
Project.....	80
Documentation Submenu.....	80
Model Validation Submenu.....	81
Import/Export Submenu.....	81
Version Control Submenu.....	82
Security Submenu.....	82
Diagram.....	83
Diagram Advanced Menu.....	84
Element.....	85
Advanced Submenu.....	86
Inline Features Submenu.....	87
Source Code Engineering Submenu.....	88
Appearance Submenu.....	88
Position Submenus.....	88
Tools	90
Source Code Engineering Submenu.....	91
Database Engineering Submenu.....	92
Web Services Submenu.....	92
XML Schema Submenu.....	92
Model Transformation (MDA) Submenu.....	93
Spelling Tools.....	93
Data Management Submenu.....	93
Manage .EAP File Submenu.....	94
Analyzer.....	94
Debug Menu.....	95
Record Menu.....	96
Test Menu	97
More Analyzer Window s.....	97
Addins.....	97
Settings.....	98
Project Types Submenu.....	99
Window	100
Help	101
Standard Windows	102
Dock Window s.....	103
Autohide Window s.....	105
Standard Toolbars	106
Default Tools Toolbar	107
Project Toolbar	108

Code Generation Toolbar	109
UML Elements Toolbar	110
Diagram Toolbar	111
Current Element Toolbar	112
Current Connector Toolbar	114
Workspace Layouts	115
Debug & Record Toolbar	116
Status Bar	117
Customization	118
The Customize Dialog	118
Customize Commands	118
Customize Toolbars	119
Custom Tools	120
Open External Tools	121
Pass Parameters to Applications	123
Customize Keyboard	124
Customize Menu	125
Customize Options	125
Visual Styles	125
Manage Workspace Layout	126
Other Windows	127
The Output Window	127
The Web Browser	128
Keyboard Shortcuts	128
Keyboard-Mouse Shortcuts	133

Part III Projects and Teams

135

Introduction	136
Projects Defined	137
Open a Project	138
Project Shortcuts	140
Create Project Shortcut	141
Capture Current Work Environment	142
Encrypt Repository Password	143
File Based Repositories	145
Copy a Base Project	146
Copy Existing Project	147
Server Based Repositories	148
Create a Repository	148
Access 2007 Repository	149
MySQL Repository	149
SQL Server Repository	150
Oracle Data Repository	151
PostgreSQL Repository	151
Adaptive Server Anywhere Repository	152
MSDE Server Repository	152
Progress OpenEdge Repository	153
Set Up an ODBC Driver for a Connection to a Repository	154
MySQL ODBC Driver	154
PostgreSQL ODBC Driver	155
ASA ODBC Driver	158
Progress OpenEdge ODBC Driver	159
Oracle ODBC Driver	160
Connect to a Data Repository	161
MySQL Data Repository	162
SQL Server Data Repository	163
Oracle Data Repository (ODBC)	164

Oracle Data Repository (OLE DB).....	166
PostgreSQL Data Repository	167
ASA Data Repository.....	169
MSDE Server Data Repository	170
Progress OpenEdge Repository.....	170
Upsizing	172
Upsize to MySQL.....	172
Upsize to Access 2007.....	174
Upsize to SQL Server	176
Upsize to Oracle 9i, 10g or 11g.....	177
Upsize to PostgreSQL.....	179
Upsize to Sybase ASA	180
Upsize to MSDE.....	182
Upsize to Progress OpenEdge.....	182
The WAN Optimizer	183
Team Development	185
Project Sharing	185
Share Enterprise Architect Projects.....	186
Refresh View of Shared Project.....	187
Share Projects on Network Drive.....	187
Distributed Development.....	188
Replication.....	188
Design Masters.....	190
Create Replicas.....	190
Synchronize Replicas.....	191
Remove Replication.....	192
Upgrade Replicas.....	193
Resolve Conflicts.....	193
Configure User Security	194
Enable Security	196
Security Policy.....	197
Maintain Users.....	198
Import User IDs From Active Directory.....	199
Assign User To Groups	201
Set Up Single Permissions.....	201
View All User Permissions.....	202
Maintain Groups	203
Set Group Permissions.....	204
List of Available Permissions	205
View and Manage Locks	207
Password Encryption.....	208
Change Password.....	209
Lock Model Elements.....	210
Add Connectors To Locked Elements.....	211
Lock Packages.....	212
Apply a User Lock.....	213
Locked Element Indicators.....	214
Identify Who Has Locked An Object.....	215
Manage Your Own Locks.....	215
Team Review Tools	216
Context Menu	218
Add a New Category	220
Add a New Topic.....	221
Add a New Post.....	222
Reply to a Post.....	224
Edit an Item.....	226
Add Object Links.....	227
Team Review Resources.....	228

Search Team Review	229
Team Review Options.....	229
Team Review Connections	230
Workflow Scripts	231
Workflow Script Functions.....	232
Functions - Validate and Control User Input	233
Functions - Create a Search With User Tasks	234
Filled Workflow Data Structures.....	234
Workflow Data Structures You Fill.....	235
Functions You Call.....	236
Sharing Reference Data	236
Export Reference Data.....	237
Import Reference Data.....	239
Change Management	241
Version Control	242
Introduction.....	243
Version Control Usage.....	244
Version Control of Model Data.....	245
Version Control and Reference data.....	246
Version Control Basics.....	246
Applying Version Control in a Team Environment.....	247
Version Control Nested Packages.....	248
Project Browser Indicators.....	249
Offline Version Control.....	250
Version Control Branching.....	251
Version Control Product Setup.....	252
System Requirements.....	253
Create a Subversion Environment.....	254
Create a new Repository Sub-tree.....	256
Create a Local Working Copy	256
Verify the SVN Workspace.....	257
Subversion Under WINE-Crossover	258
Preparing a Subversion Environment Under Wine.....	259
TortoiseSVN.....	260
Create a CVS Environment.....	260
Prepare a CVS Local Workspace.....	262
Verify the CVS Workspace.....	263
TortoiseCVS.....	264
Create a TFS Environment.....	264
TFS Workspaces	265
TFS Exclusive Check Outs.....	266
Create an SCC Environment.....	267
Version Control Setup.....	268
Re-use an Existing Configuration.....	269
Version Control Settings Dialog.....	270
SCC Settings.....	271
CVS Settings.....	272
SVN Settings.....	273
TFS Settings.....	274
Use Version Control.....	275
Project Version Control Menu.....	277
Package Version Control Menu.....	278
Configure Controlled Package.....	282
Check In a Model Branch.....	283
Check Out a Model Branch.....	284
Check In a Package.....	285
Check Out a Package.....	285
Undo Check Out of a Package.....	286

Update to the Latest Revision of Selected Package.....	287
Update to the Latest Revision of All Packages.....	288
Include Other Users' Packages.....	289
Apply Version Control To Branches.....	289
Export Controlled Model Branch.....	290
Import Controlled Model Branch.....	291
Manually Locating Model Branch Files.....	292
Review Package History.....	293
Review Package History - SCC Client.....	294
Retrieve Prior Revision - SCC Client.....	295
Validate Package Configurations.....	296
Resynchronize the Status of Version Controlled Packages.....	297
Tracking Changes	298
Auditing.....	298
Auditing Quickstart.....	300
Auditing Settings.....	300
The Audit View	302
Audit View Controls.....	304
Audit History Tab.....	307
Auditing Performance Issues.....	307
Audit View Performance Issues.....	308
Package Baselines.....	308
Baselines	310
Manage Baselines.....	311
Create Baselines.....	312
The Compare Utility (Diff).....	313
Compare Options	314
Example Comparison.....	315
Compare Utility Tab Options.....	316
Model Transfer	318
XML Import and Export.....	319
Export to XML.....	320
Publish Model Package.....	321
Import from XML.....	323
Import EMX/UML2 Files.....	324
Limitations of XML.....	325
The UML DTD.....	326
Controlled Packages	327
Controlled Package Menu.....	328
Configure Packages.....	329
Remove Package from Control.....	331
Save a Package.....	331
Load a Package	332
Batch XML Export.....	332
Batch XML Import.....	333
Manual Version Control with XML.....	334
Report Deletion of Cross Package References.....	335
CSV Import and Export.....	337
Compare Projects.....	337
CSV Specifications.....	338
Using Preserve Hierarchy.....	340
CSV Export.....	340
CSV Import.....	342
Perform a Project Data Transfer	343
Copy Packages Between Projects	344
Project Management	346
The Project Management Window	347
Project Resources	348

Resource Allocation.....	349
Effort Management.....	350
Risk Management.....	351
Metrics.....	352
Effort Types.....	353
Metric Types.....	354
Risk Types.....	355
The Project Information Window	356
Project Tasks.....	356
Add, Modify and Delete Tasks.....	357
Project Issues.....	358
Add, Delete and Modify Issues.....	358
Report From Project Issues Dialog.....	360
Report From Project Issues Tab.....	360
Report Output Sample.....	361
Project Glossary.....	362
Project Glossary Tab.....	362
The Glossary Dialog.....	363
Generate a Report.....	364
Glossary Report Output Sample.....	365
Project Task Allocation	366
Resource View.....	366
Element View.....	368
Report View.....	370
Spell Checking	372
Select Spell Checker Options.....	373
Use Languages Other Than English.....	375
Using the Spell Checker.....	376
Correcting Words.....	377
Personal Information	377
Internal Mail.....	378
Create a Message.....	381
Recording Work.....	382
Monitor Your Tasks.....	385
Monitor Workflow.....	386
Working Sets.....	387
Project Calendar	390
Calendar.....	392
Configure Event Subtypes.....	395
Allocated Resources.....	396
Project Tasks.....	398
Use Case Estimation	400
Technical Complexity Factors.....	401
Environment Complexity Factors.....	402
Default Hours.....	404
Estimating Project Size.....	404
SPEM	406
SPEM Toolbox Pages.....	407
Update Package Status	411
Manage Bookmarks	412
Monitor Change Events	413
Project Maintenance	414
Check Project Data Integrity.....	414
Reset Table Auto Increment or Identity Columns.....	416
Project Upgrade.....	417
Upgrade Replicas.....	417
Run SQL Patches.....	418
Rename a Project.....	418

Compact a Project	418
Repair a Project	419
Local Options	421
General	422
Standard Colors	424
Create Project Colors	425
Diagram	426
Appearance	427
Set Default Fonts	429
Behavior	430
Sequence	431
Objects	432
Element Visibility	434
Links	434
Communication Message Colors	435
XML Specifications	436
Part IV Navigate, Search & Trace	440
Navigate: Exploring Your Model	441
The Project Browser	441
Project Browser Context Menus	443
Model (Root Node) Context Menu	443
Package Menu	444
Add Sub-Menu	446
Documentation Sub-Menu	447
Code Engineering Sub-Menu	447
Execution Analyzer Sub-Menu	448
Import/Export Sub-Menu	448
Contents Sub-Menu	448
Element Menu - Project Browser	448
Add Sub Menu	450
Diagram Menu - Project Browser	451
Operation Menu - Project Browser	452
Project Browser Toolbar	452
Project Browser Icon Overlays	454
Order Package Contents	455
Set Default Behavior	455
Package Browser	456
List Header	458
Package Browser Options	459
Diagram List	462
Model Views	464
Model Views Toolbar	466
Model Views Context Menus	467
Model Views Operations	469
Diagram Slide Show	471
The Pan & Zoom Window	473
Search: Finding Information	475
Model Search	475
Model Search Context Menu	477
Model Search Toolbar	478
Pre-defined Searches	479
Customizing the Search View	480
Create & Modify Searches	482
Create Search Definitions	484
Advanced Search Options	486
Add Filters	487

Fields and Conditions	489
Diagram Filters	489
Work With Diagram Filters	491
Trace: Tracking Dependencies	493
Traceability Tools	494
The Traceability Window	495
Relationship Matrix	496
Open the Relationship Matrix	497
Set Element Type	497
Set Connector Type and Direction	498
Set Source and Target Package	498
Matrix Profiles	499
Creating and Deleting Relationships	500
Relationship Matrix Options	501
Review Source and Target Elements	503
The Relationships Window	504
Example Diagram	505
Packages and Elements	506
Gap Analysis Matrix	510
Edit Gap Notes	512

Part V Modeling Basics **515**

Modeling	517
Models	518
Model Wizard	518
Model Templates	519
Business Process Model Template	520
Requirements Model Template	521
Use Case Model Template	521
Domain Model Template	522
Class Model Template	523
Database Model Template	524
Component Model Template	525
Deployment Model Template	526
Testing Model Template	527
Maintenance Model Template	528
Project Model Template	528
Views	530
Add Views	531
Rename Views	531
Delete Views	532
Packages	533
Open Package in the Project Browser	533
Add a Package	534
Rename a Package	534
Copy a Package	535
Drag a Package Onto a Diagram	536
Show or Hide Package Contents	537
Delete a Package	537
Diagrams	538
Diagram Context Menu	538
Insert Elements and Connectors	540
Print Preview	540
Diagram View	540
Format Toolbar	541
Diagram Tabs	543

View Tab Switcher.....	545
Diagram Toolbox	546
Toolbox Appearance Options	549
Toolbox Shortcut Menu.....	551
Common Group.....	552
Use Case Group.....	553
Class Group.....	554
Object Group.....	555
Composite Group.....	555
Communication Group.....	556
Interaction Group.....	556
Timing Group.....	557
State Group.....	558
Activity Group.....	558
Component Group.....	559
Deployment Group.....	559
Profile Group.....	560
Metamodel Group.....	561
Analysis Group.....	561
Custom Group.....	562
Requirement Group.....	563
Maintenance Group.....	564
User Interface Group.....	564
WSDL Group.....	565
XML Schema Group.....	566
Data Modeling Group.....	566
Test Domain Group.....	567
Diagram Tasks	567
Add New Diagrams.....	568
Diagram Properties.....	569
General Tab.....	570
Diagram Tab.....	571
Elements Tab.....	573
Features Tab.....	574
Visible Class Members.....	575
Connectors Tab.....	576
Paste from Project Browser.....	576
Paste Multiple Items.....	578
Paste Composite Elements.....	578
Paste Activities.....	579
Copy And Paste Diagram Element.....	579
Place Related Elements on Diagram.....	580
Delete Diagram.....	580
Rename Diagram.....	581
Change Diagram Type.....	581
Diagram Navigation Hotkeys.....	582
Copy Image to Disk.....	582
Copy Image to Clipboard.....	583
Copy (Duplicate) Diagram.....	583
Z Order Elements.....	584
Set the Default Diagram.....	585
Open Package From Diagram.....	585
Feature Visibility.....	585
Insert Diagram Properties Note.....	587
Create Legends.....	588
Legend Properties.....	588
Style Options.....	589
Autosize Elements.....	589

Swimlanes.....	590
Swimlane Details.....	591
Swimlanes Matrix.....	592
Using the Image Manager.....	593
Create Custom Diagram Background.....	595
Import Image Library.....	595
Show Realized Interfaces of Class.....	597
Label Menu Section.....	597
Pan and Zoom a Diagram.....	599
Move Elements In Diagram Sections.....	600
View Last and Next Diagram.....	600
Set Diagram Page Size.....	600
Scale Image to Page Size.....	602
Lock Diagram.....	602
Undo Last Action.....	603
Redo Last Action.....	603
Layout Diagrams	604
Circular/Elliptical Layout.....	605
Box Layout.....	609
Per Page Layout.....	611
Digraph Layout.....	612
Spring Layout.....	613
Neaten Layout.....	614
Converge/Diverge Layout.....	615
Fan Relations Layout.....	617
Auto Route Layout.....	618
Lay Out a Diagram Automatically	619
The Quick Linker	622
Create New Elements.....	623
Create Connectors.....	624
Elements	625
Element Tasks	625
Create Elements.....	626
Add Elements Directly To Packages.....	627
Use Auto Naming and Auto Counters	628
Set Element Parent.....	628
Show Element Use.....	630
Set Up Cross References.....	630
Move Elements Within Diagrams.....	632
Move Elements Between Packages.....	633
Copy Elements Between Packages.....	634
Change Element Type.....	635
Align Elements.....	635
Resize Elements.....	636
Delete Elements.....	637
Customize Visibility of Elements.....	638
Create Notes and Text.....	639
Link Note to Internal Documentation.....	640
Set an Element's Default Appearance.....	641
Get/Set Project Custom Colors.....	642
Set Element Templates Package.....	643
Highlight Context Element.....	645
Make Linked Element a Local Copy	646
Copy Features Between Elements.....	646
Move Features Between Elements	647
Composite Elements	647
Element Context Menu	648
Properties Menu Section.....	649

Advanced Submenu.....	649
Add Submenu.....	651
Insert Related Elements.....	652
Find Submenu.....	653
Embedded Elements Submenu.....	653
Embedded Elements Window.....	654
Features Menu Section.....	654
Code Engineering Menu Section.....	655
Appearance Menu Section.....	655
Set Element Font.....	657
Element Multiple Selection Menu.....	658
Visual Representation	658
Element Icons.....	659
Compartment.....	659
Element Property Displays	660
Properties Dialog.....	660
General Settings.....	662
Details.....	663
Advanced Properties.....	664
Templates.....	664
Requirements.....	664
External Requirements.....	665
Constraints.....	666
Links.....	666
Scenarios.....	667
Structured Specification Tab.....	669
Set Up Scenario Specification.....	669
Structured Specification Toolbar.....	671
Structured Specification Item Context Menu.....	672
Structured Specification Selected Text Context Menu.....	674
Structured Specification Entry Points Context Menu.....	675
Structured Specification Floating Toolbar.....	676
Generate Diagrams.....	676
Generated Activity Diagram.....	677
Generated RuleFlow Diagram.....	678
Generated State Machine Diagram.....	679
Generated Sequence Diagram.....	681
Generated Robustness Diagram.....	683
Generate Scenario From Activity Diagram.....	684
Generate Test Cases.....	685
Context References Tab.....	686
Scenario Constraints Tab.....	686
Associated Files.....	687
The Element Browser.....	687
Properties Window.....	689
The Scenarios & Requirements Window.....	689
Select <Item> Dialog.....	690
Select Property Dialog.....	692
Set Feature Dialog.....	694
Attributes	695
Attributes Dialog - General.....	696
Attributes Dialog - Detail.....	698
Attributes Dialog - Constraints.....	699
Attribute Tagged Values.....	699
Create Properties.....	700
C# Automatic Property Support.....	701
Display Inherited Attributes.....	702
Create Object From Attribute.....	704

Object Classifiers	705
Using Classifiers	706
Behavior	706
Operations.....	707
Operations Dialog - General.....	708
Operations Dialog - Behavior.....	710
Initial Code.....	711
Operations Dialog - Constraints.....	711
Operation Tagged Values.....	712
Override Parent Operations.....	712
Display Inherited Operations.....	713
Interactions and Activities.....	715
Behavior Calls	715
Associate with Different Behaviors	716
Synchronize Arguments.....	716
Behavior Call Arguments.....	716
Behavior Parameters.....	717
Parameters Dialog.....	717
Parameter Tagged Values	718
Operation Parameters by Reference.....	719
In-place Editing Options	720
In-place Editing Tasks.....	720
Edit Element Item Name.....	722
Edit Feature Stereotype.....	722
Edit Feature Scope.....	723
Edit Attribute Keyw ord.....	724
Edit Operation Parameter Keyw ord.....	724
Edit Parameter Kind.....	725
Insert New Feature	726
Insert Operation Parameter	726
Insert Maintenance Feature.....	727
Insert Testing Features	728
Linked Documents	729
Create Document Artifact.....	730
Link Document to UML Element.....	730
Edit Linked Documents	731
Hyperlink From Linked Document	732
Create Element From Document.....	732
Replace or Delete Documents.....	733
Create Linked Document Templates.....	734
Edit Linked Document Templates.....	734
Connectors	736
Connector Context Menu	736
Properties Menu Section.....	737
Type-Specific Menu Section.....	737
Advanced Menu Section.....	738
Style Menu Section.....	739
Appearance Menu Section.....	740
Connector Tasks	740
Connect Elements.....	741
Connect to Element Feature.....	742
Add a Note to a Connector.....	743
Arrange Connectors	744
Change Connector Type.....	744
Change the Source or Target Element.....	745
Connector Styles.....	745
Create Connector in Project Brow ser.....	748
Relationship Visibility.....	749

Delete Connectors.....	750
Generalization Sets.....	750
Hide/Show Connectors.....	751
Hide/Show Labels.....	752
Connector In-place Editing Options.....	753
Reverse Connector.....	753
Set Association Specializations.....	753
Change Sequence Message Scope.....	754
Show Uses Arrow Head.....	754
Tree Style Hierarchy.....	755
Connector Properties	756
Connector Constraints.....	757
Binding.....	758
Source Role.....	758
Target Role.....	760
Connector Tagged Values.....	760
Connector Advanced.....	761
Tagged Values	762
Quick Add - Tagged Value To Elements	763
Assign a Tagged Value to an Item	764
Assign Notes to a Tagged Value	765
Modify Tagged Values	766
Show Duplicate Tags	766
Advanced Tag Management	767
Notes	769
Notes Toolbar	770
Reference Data	772
UML Types	772
Stereotype Settings.....	773
Shape Editor.....	774
Tagged Value Types.....	775
Cardinality.....	776
Data Types.....	777
People	778
Project Authors.....	778
Select Users.....	779
Project Roles.....	780
Project Resources.....	781
Project Clients.....	782
General Types	782
Status Types.....	783
Constraint Types.....	784
Constraint Status Types.....	785
Requirement Types.....	785
Scenario Types.....	786
Metrics and Estimation.....	787
Maintenance	787
Problem Types.....	787
Testing Types.....	788
Resources	789
Favorites.....	790
Part VI Standard UML Models	793
UML Diagrams	795
UML Structural Models	796
Package Diagram	796

Example Package Diagram.....	797
Class Diagram	798
Example Class Diagram.....	799
Object Diagram	799
Example Object Diagram.....	800
Composite Structure Diagram	801
Example Composite Structure Diagram.....	802
Properties.....	803
Deployment Diagram	804
Example Deployment Diagram.....	806
Component Diagram	807
Example Component Diagram.....	808
Profile Diagram	809
Example Profile Diagram.....	810
UML Behavioral Models	811
Activity Diagram	811
Example Activity Diagram.....	813
Use Case Diagram	813
Example Use Case Diagram.....	815
State Machine Diagrams	815
Example State Machine Diagram.....	817
Pseudo-States.....	820
Regions.....	820
State Machine Table	821
State Machine Table Options.....	822
State Machine Table Operations	823
Change State Machine Table Position.....	824
Change State Machine Table Size.....	824
Insert New State.....	824
Insert Trigger.....	825
Insert/Change Transition.....	825
Reposition State or Trigger Cells.....	826
Add Legend.....	826
Find Cell in State Machine Diagram.....	827
State Machine Table Conventions.....	828
Export State Table To CSV File.....	828
Example State-Trigger Table.....	829
Example State-Next State Table.....	830
Timing Diagram	830
Example Timing Diagram.....	831
Create a Timing Diagram.....	832
Set a Time Range.....	832
Edit a Timing Diagram.....	833
Add and Edit State Lifeline.....	833
Add States to a State Lifeline.....	834
Edit States in a State Lifeline.....	835
Delete States in a State Lifeline.....	835
Edit Transitions In State Lifeline.....	835
Add and Move Transitions.....	836
Add and Edit Value Lifeline.....	838
Add States In Value Lifeline.....	838
Edit Transitions In Value Lifeline.....	838
Configure Timeline - States.....	840
Numeric Range Generator	841
Configure Timeline - Transitions.....	842
Time Intervals	843
Create Time Intervals.....	843
Compress Time Intervals.....	845

Select Time Intervals.....	846
Time Interval Operations.....	847
Sequence Diagram	849
Example Sequence Diagram.....	851
Denote Lifecycle of an Element.....	851
Layout of Sequence Diagrams.....	852
Sequence Elements.....	853
Sequence Diagrams and Version Control.....	854
Sequence Element Activation.....	855
Lifeline Activation Levels.....	856
Sequence Message Label Visibility.....	858
Change the Top Margin.....	858
Inline Sequence Elements.....	858
Communication Diagram	859
Example Communication Diagram.....	860
Communication Diagrams in Color.....	861
Interaction Overview Diagram	861
Example Interaction Overview Diagram.....	862
UML Elements	864
Behavioral Diagram Elements	864
Action.....	865
Action Notation.....	866
Action Expansion Node.....	868
Action Pin	869
Assign Action Pins.....	870
Local Pre/Post Conditions	871
Class Operations in Activity Diagrams.....	872
Activity.....	873
Activity Notation.....	874
Activity Parameter Nodes	875
Activity Partition.....	876
Actor.....	877
Central Buffer Node.....	878
Choice.....	879
Combined Fragment.....	880
Create a Combined Fragment.....	882
Interaction Operators.....	882
Datastore.....	885
Decision.....	886
Diagram Frame.....	888
Diagram Gate.....	889
Endpoint.....	890
Entry Point.....	891
Exception.....	892
Expansion Region.....	893
Exit Point.....	895
Final	895
Flow Final.....	897
Fork/Join.....	898
Fork	900
Join	901
History.....	902
Initial	903
Interaction.....	904
Interaction Occurrence.....	906
Interruptible Activity Region.....	907
Junction.....	908
Lifeline.....	910

Merge.....	910
Message Endpoint.....	911
Message Label.....	912
Note	913
Partition.....	914
Receive.....	915
Region.....	916
Send.....	917
State.....	917
Composite State.....	919
State/Continuation.....	920
Continuation.....	921
State Invariant.....	922
State Lifeline.....	923
State Machine.....	925
Structured Activity.....	926
Structured Node.....	927
Sequential Node.....	927
Loop Node.....	928
Conditional Node.....	929
Synch.....	930
System Boundary.....	931
Boundary Element Settings.....	932
Terminate.....	933
Trigger.....	934
Use Case.....	935
Use Case Extension Points.....	937
Rectangle Notation.....	938
Value Lifeline.....	938
Structural Diagram Elements	940
Artifact.....	940
Create Artifact For External File.....	941
Class.....	941
Active Classes.....	943
Parameterized Classes (Templates).....	943
Collaboration.....	945
Collaboration Use.....	946
Component.....	948
Data Type.....	948
Deployment Spec.....	949
Device.....	950
Document Artifact.....	951
Enumeration.....	952
Execution Environment.....	952
Expose Interface.....	953
Information Item.....	954
Interface.....	955
Node.....	956
Object.....	957
Run-time State.....	958
Define a Run-time Variable.....	958
Remove a Defined Variable.....	959
Object State.....	959
Package.....	960
Part	960
Add Property Value.....	961
Port	962
Add a Port to an Element.....	962

Inherited and Redefined Ports.....	963
The Property Page.....	963
Primitive.....	964
Signal.....	964
UML Connectors	966
Aggregation	968
Change Aggregation Connector Form.....	969
Assembly	969
Association	970
Qualifiers.....	971
Qualifiers Dialog.....	972
Association Class	973
Connect New Class to Association.....	975
Communication Path	975
Composition	976
Connector	977
Control Flow	977
Delegate	978
Dependency	979
Apply a Stereotype.....	979
Deployment	980
Extend	980
Generalization	981
Include	982
Information Flow	983
Using Information Flow s.....	983
Convey Information on a Flow	984
Realize an Information Flow	985
Interrupt Flow	985
Manifest	986
Message	986
Message (Sequence Diagram).....	987
Self-Message.....	989
Call	991
Message Examples.....	992
Change the Timing Details.....	993
General Ordering.....	994
Asynchronous Signal Message.....	996
Co-Region Notation.....	996
Message (Communication Diagram).....	996
Create a Communication Message.....	997
Re-Order Messages	998
Message (Timing Diagram).....	999
Create a Timing Message.....	1000
Nesting	1002
Notelink	1003
Object Flow	1003
Object Flow s in Activity Diagrams	1004
Occurrence	1005
Package Import	1006
Package Merge	1006
Realization	1007
Recursion	1008
Role Binding	1008
Represents	1009
Representation	1010
Template Binding	1010
Parameter Substitution.....	1011

Trace	1012
Transition	1013
Use	1014
UML Stereotypes	1016
Apply Stereotypes	1017
Stereotype Selector	1017
Stereotype Visibility	1018
Standard Stereotypes	1019
Stereotypes with Alternative Images	1020
UML Patterns	1021
Create a Pattern	1022
Import a Pattern	1023
Use a Pattern	1024
Add Pattern Dialog	1025
UML Profiles	1026
Import a UML Profile	1027
Add Profile Objects to a Diagram	1028
Tagged Values in Profiles	1028
Synchronize Tagged Values and Constraints	1029
MDG Technologies	1031
Work with MDG Technologies	1031
Manage MDG Technologies	1033
Access Remote MDG Technologies	1034
Import MDG Technologies to APPDATA	1035
Extensions - MDG Technologies	1036
Define a Modeling Language	1038
MDG Technology SDK	1038
Developing Profiles.....	1039
Custom Stereotypes.....	1039
Create Profiles.....	1041
Create a Profile Package.....	1041
Add Stereotypes and Metaclasses.....	1042
Define Stereotype Tagged Values.....	1044
With Predefined Tag Types.....	1045
With Supported Attributes.....	1045
Use the Tagged Value Connector.....	1046
Define Stereotype Constraints.....	1046
Add Enumeration Elements.....	1047
Add Shape Scripts.....	1048
Set Default Appearance.....	1049
Export a Profile.....	1050
Save Profile Options.....	1050
Supported Attributes.....	1051
Define a Stereotype as a Metatype.....	1054
Define Multiple-Stereotype Level.....	1054
Define Creation of Instance.....	1055
Create Composite Elements.....	1056
Define Child Diagram Types.....	1057
Stereotype Profiles.....	1058
Quick Linker.....	1058
Quick Linker Definition Format.....	1059
Quick Linker Example.....	1061
Hide Default Quick Linker Settings.....	1062
Quick Linker Object Names.....	1062
MDG Technologies - Creating.....	1064
Create MDG Technologies.....	1064

Add a Profile.....	1066
Add a Pattern.....	1067
Add a Diagram Profile.....	1068
Add a Toolbox Profile.....	1068
Add Learning Center Pages.....	1069
Add Tagged Value Types.....	1069
Add Code Modules.....	1070
Add MDA Transforms.....	1071
Add Images.....	1071
Add Scripts.....	1072
Add RTF Report Templates.....	1072
Add Linked Document Templates.....	1072
Working with MTS Files.....	1073
Customize Toolbox Profiles.....	1074
Create Toolbox Profiles.....	1074
Toolbox Page Attributes.....	1075
Create Hidden Sub-Menus.....	1075
Override Default Toolboxes.....	1076
Assign Icons To Toolbox Items.....	1076
List of Enterprise Architect Toolboxes.....	1077
Elements Used in Toolboxes.....	1077
Connectors Used In Toolboxes.....	1079
Create Custom Diagram Profiles.....	1080
Built-In Diagram Types.....	1081
Attribute Values - StyleEx & pdata.....	1081
Create Learning Center Profiles.....	1082
Define Learning Center Toolboxes.....	1083
Built-In Learning Center Commands.....	1084
Run Add-In Functions.....	1084
Define Learning Center Contexts.....	1085
Allocate Learning Center Contexts.....	1086
Save a Learning Center Profile.....	1086
Define Validation Configuration.....	1086
Incorporate Model Templates.....	1087
Set Up Technology Element Images.....	1088
Deploy An MDG Technology.....	1089
Shape Scripts.....	1089
Getting Started With Shape Scripts.....	1090
Shape Editor.....	1090
Write Scripts.....	1091
Syntax Grammar.....	1091
Shape Attributes.....	1092
Drawing Methods.....	1094
Color Queries.....	1099
Conditional Branching.....	1099
Query Methods.....	1099
Display Element/Connector Properties.....	1099
Sub-Shapes.....	1102
Reserved Names.....	1103
Miscellaneous.....	1103
Example Scripts.....	1104
Tagged Value Types.....	1108
Predefined Structured Types.....	1108
Create Structured Tagged Values.....	1111
Predefined Reference Data Types.....	1111
Create Reference Data Tagged Values.....	1113
Create Custom Tagged Value Type.....	1113
Code Template Framework.....	1114

Code Template Syntax.....	1115
Literal Text.....	1115
Macros	1115
Template Substitution Macros.....	1116
Field Substitution Macros.....	1116
Tagged Value Substitution Macros.....	1129
Function Macros.....	1130
Control Macros.....	1133
EASL Code Generation Macros.....	1137
EASL Collections.....	1138
EASL Properties.....	1141
Variables.....	1145
The Code Template Editor in MDG Development.....	1147
Custom Templates.....	1147
Override Default Templates.....	1148
Add New Stereotyped Templates.....	1149
Create Custom Language Template.....	1150

Part VII Requirement Models 1152

Requirements	1154
Create Requirements	1156
Requirement Properties	1157
Color Code External Requirements	1159
Extend Requirement Properties	1160
Display Tagged Values On Diagrams	1161
Connect Requirements	1161
Import Requirements and Hierarchies in CSV	1162
Model Requirements	1164
Feature	1166
Internal Requirements	1167
Make Internal Requirement External.....	1167
Manage Requirements	1169
View Requirements	1170
Trace Use of Requirements	1170
Manage Requirement Changes	1171
Report on Requirements	1173
Requirements Diagram	1174
Example Requirements Diagram	1174

Part VIII Domain Based Models 1178

Domain Based Diagrams	1181
Analysis Models	1182
Mind Mapping	1182
Custom Diagram	1183
Data Flow Diagrams	1184
Analysis Stereotypes	1186
Analysis Diagram	1187
Example Analysis Diagram.....	1189
Business Models	1190
Business Modeling/Interaction	1191
Example Business Modeling Diagram.....	1192
Business Models	1193
Analysis Models.....	1195
Process Modeling Notation.....	1195
Inputs, Resources and Information.....	1196

Events	1197
Outputs	1197
Goals	1198
A Complete Business Process	1198
Business Rules	1199
Model Business Rules For RuleTasks	1202
Create a Business Domain Model	1204
Create a Rule Flow Model	1206
Pass Parameters to Rule Flow Activity	1208
Model Rules In an Operation	1209
Compose Business Rules	1211
Validate Business Rules	1216
Code Generation For Business Rules	1217
BPMN Models	1219
BPMN 1.0 and 1.1 Toolbox Pages	1220
BPMN 2.0 Toolbox Pages	1222
BPMN 2.0 Business Process Toolbox Pages	1222
BPMN 2.0 Choreography Toolbox Pages	1224
BPMN 2.0 Conversation Toolbox Pages	1225
BPMN 2.0 Type Toolbox Page	1227
Change BPMN Element Appearance	1228
Migrate BPMN 1.0 Model to BPMN 1.1	1232
Migrate BPMN 1.1 Model to BPMN 2.0	1233
BPMN 2.0 XML	1234
BPEL Models	1235
BPEL 1.1 Model	1236
Create BPEL 1.1 Model Structure	1238
Model a BPEL 1.1 Process	1240
Start Event	1240
Intermediate Event	1242
Activity	1244
Gateway	1246
End Event	1247
Sequence Flow	1248
Pool	1249
Assignment	1250
Create BPEL 1.1 Web Service	1251
Generate BPEL 1.1	1253
BPEL 2.0 Model	1254
Create BPEL 2.0 Model Structure	1256
Model a BPEL 2.0 Process	1257
Start Event	1258
Intermediate Event	1259
Activity	1261
Gateway	1263
End Event	1264
Data Object	1265
Property	1266
Sequence Flow	1266
Pool	1267
Assignment	1267
Create BPEL 2.0 Web Service Operation	1268
Generate BPEL 2.0	1271
BPEL Model Validation	1271
ArchiMate	1272
Eriksson-Penker Extensions	1273
Data Models	1276
Table	1277

Database Schema	1278
Conceptual Data Model	1278
Logical Data Model	1279
User Interface Models	1281
Screen	1281
UI Control Elements	1282
Web Stereotypes	1284
Example User Interface Diagram	1286
Win32 UI Technology	1286
Other Stereotypes	1288
Boundary	1288
Create a Boundary	1289
Control	1289
Create a Control Element	1290
Entity	1290
Create an Entity	1291
Event	1292
Hyperlinks	1292
Hyperlinks To Files	1293
Script Hyperlinks	1294
Add Action As Hyperlink	1294
Hyperlinks Between Diagrams	1294
Image	1295
N-Ary Association	1296
Process	1297
Packaging Component	1297
Risk	1298
Task	1299

Part IX Model Transformation

1301

Transform Elements	1305
Chaining Transformations	1306
Built-in Transformations	1307
C# Transformation	1307
Data Model To ERD Transformation	1310
DDL Transformation	1310
EJB Transformations	1314
ERD To Data Model Transformation	1317
Java Transformation	1319
JUnit Transformation	1321
NUnit Transformation	1323
WSDL Transformation	1325
XSD Transformation	1326
Transformation Templates	1330
Import Transformations	1331
Write Transformations	1332
Default Transformation Templates	1332
Intermediary Language	1333
Objects	1333
Connectors	1338
Copy Information	1340
Convert Types	1340
Convert Names	1341
Cross References	1342

Part X Database Engineering 1345

Physical Data Model 1346

Tables and Columns 1347

 Create a Table..... 1347

 Working with Table Properties..... 1348

 Set the Database Type..... 1348

 Set Table Owner/Schema..... 1349

 Set MySQL Options..... 1349

 Set Oracle Table Properties..... 1350

 Create Columns..... 1351

 Reorder Columns 1352

Data Types 1353

 Add New Datatypes..... 1353

 MySQL Data Types..... 1354

 Oracle Data Types..... 1354

 Map Data Types between DBMS products..... 1355

 Data Type Conversion Procedure..... 1356

 Data Type Conversion for a Package..... 1356

Database Keys 1357

 Primary Key..... 1358

 Create a Primary Key..... 1359

 Define Primary Key Name Template..... 1359

 SQL Server Non Clustered Keys..... 1360

 Foreign Key..... 1361

 Create a Foreign Key..... 1361

 Composite Foreign Key..... 1364

 Define Foreign Key Name Template..... 1365

Stored Procedures 1366

Indexes 1366

 Create a Stored Procedure..... 1367

 Create an Index..... 1367

Triggers 1369

 Create a Trigger..... 1369

Check Constraints 1370

 Create a Check Constraint..... 1370

Views 1370

 Create a View 1371

Oracle Packages 1373

Default Constraints 1373

A Data Model Diagram 1374

Import Database Schema 1375

 Select a Data Source 1377

 Select Tables 1377

 The Imported Class Elements 1377

Generate DDL 1379

 Generate DDL For a Table 1379

 Generate DDL for a Package 1380

Entity Relationship Diagrams (ERDs) 1382

Part XI Software Engineering 1386

Development Tools 1387

 Overview of Development 1389

Code, Build & Debug 1390

 Getting Started 1391

Prerequisites	1391
Available Tools	1391
Workspace Layout	1391
General Workflow	1391
Code Generation and Synchronization - Safeguards	1392
Setup	1392
Managing Analyzer Scripts	1392
Analyzer Script Editor	1394
Code Editors	1394
Syntax Highlighting	1395
Bookmarks	1396
Cursor History	1396
Brace Matching	1397
Automatic Indentation	1397
Commenting Selections	1397
Scope Guides	1398
Zooming	1398
Line Selection	1399
Intellisense	1399
Code Editor Key Bindings	1400
Code Editor Context Menu	1404
Create Use Case for Method	1406
Script Editor	1407
The Source Code Viewer	1408
Source Code Viewer Toolbar	1410
Find and Replace	1412
Build	1415
Add Commands	1415
Recursive Builds	1416
Clean	1416
Add Clean Commands	1416
Debugging	1416
How it Works	1416
Setup for Debugging	1417
Operating System Specific Requirements	1417
UAC-Enabled Operating Systems	1418
WINE Debugging	1419
Microsoft C++ and Native (C, VB)	1420
General Setup	1421
Debug Symbols	1421
Java	1422
General Setup for Java	1422
Advanced Techniques	1423
Attach to Virtual Machine	1424
Internet Browser Java Applets	1424
Working with Java Web Servers	1425
JBoss Server	1427
Apache Tomcat Server	1428
Apache Tomcat Windows Service	1428
.NET	1428
General Setup for .NET	1429
Debugging an Unmanaged Application	1429
Debug COM Interop	1430
Debug ASP .NET	1431
The PHP Debugger	1432
PHP Debugger - System Requirements	1433
PHP Debugger Checklist	1434
Tracepoint Output	1435

Breakpoint and Marker Management.....	1436
Breakpoint Properties.....	1437
Setting Code Breakpoints.....	1438
Setting Data Breakpoints.....	1438
Failure to Bind Breakpoint.....	1439
Delete and Disable Breakpoints.....	1439
Trace Statements.....	1440
How Markers are Stored.....	1441
Debugging Actions.....	1441
View the Call Stack.....	1441
View the Local Variables.....	1442
View Content Of Long Strings.....	1443
View Elements of Array.....	1443
View Variables in Other Scopes.....	1444
Variable Snapshots.....	1444
Inspect Process Memory.....	1445
Break When a Variable Changes Value.....	1445
Show Loaded Modules.....	1446
Show Output from Debugger.....	1446
Debug Tooltips in Code Editors.....	1447
Start & Stop Debugger.....	1447
Debug Another Process.....	1448
Step Over Lines of Code.....	1448
Step Into Function Calls.....	1449
Step Out of Functions.....	1449
Recording Actions.....	1449
Step Through Function Calls.....	1449
Create Sequence Diagram of Call Stack.....	1450
Saving the Call Stack.....	1450
Run	1451
Add Run Command.....	1451
Testing	1451
Add Testing Command.....	1451
Testpoints Output.....	1453
Deploying	1453
Add Deploy Command.....	1453
Searching Files	1454
Find in Files.....	1454
Modeling Conventions	1457
ActionScript Conventions	1458
Ada 2005	1459
C Conventions	1461
Object Oriented Programming In C.....	1462
C# Conventions	1463
C++ Conventions	1465
Managed C++ Conventions.....	1467
C++/CLI Conventions.....	1468
Delphi Conventions	1469
Java Conventions	1470
AspectJ Conventions.....	1472
PHP Conventions	1472
Python Conventions	1473
System C Conventions	1474
VB.Net Conventions	1475
Verilog Conventions	1477
VHDL Conventions	1478
Visual Basic Conventions	1480
Code Template Framework	1482

Code Templates	1482
Base Templates.....	1483
The Code Template Editor	1485
Synchronize Code	1486
Synchronize Existing Sections.....	1487
Add New Sections.....	1488
Add New Features and Elements	1488
Generate Source Code	1490
Generate a Single Class	1491
Generate a Group of Classes	1492
Generate a Package	1493
Update Package Contents	1494
Namespaces	1495
Generate From Behavioral Models	1496
SW Code Generation - State Machine Diagrams	1497
Java Code Generated From State Machine Diagram.....	1498
State Machine Modeling For HDLs	1502
Code Generation - Interaction Diagrams	1503
Code Generation - Activity Diagrams	1504
Synchronize Model and Code	1506
Importing Source Code	1508
Import Source Code	1510
Notes on Source Code Import	1510
Import a Directory Structure	1513
Import Binary Module	1513
MDG Integration and Code Engineering	1514
Classes Not Found During Import	1515
Other Settings	1516
Source Code Engineering	1516
Source Code Options.....	1517
Import Component Types	1518
Options - Code Editors	1519
Editor Language Properties.....	1520
Options - Object Lifetimes.....	1521
Options - Attribute/Operations	1522
Code Page for Source Editing.....	1523
Local Paths	1523
Local Paths Dialog	1524
Language Macros	1525
Set Collection Classes	1526
Example Use of Collection Classes.....	1528
Language Options	1529
ActionScript Options	1530
Ada 2005 Options.....	1531
C Options.....	1531
C# Options.....	1532
C++ Options.....	1533
Delphi Options	1533
Delphi Properties.....	1534
Java Options	1535
PHP Options.....	1536
Python Options.....	1536
SystemC Options.....	1537
VB.NET Options.....	1537
Verilog Options.....	1538
VHDL Options.....	1538

Visual Basic Options.....	1539
MDG Technology Language Options.....	1540
Reset Options.....	1540
ICONIX	1542
GoF Patterns	1545

Part XII Systems Engineering 1548

SysML	1552
SysML Activity	1553
SysML Block Definition	1556
SysML Interaction	1558
SysML Internal Block	1559
SysML Model Elements	1561
SysML Parametrics	1563
SysML Requirements	1564
SysML State Machine	1565
SysML Use Case	1567
SysML Parametric Models	1568
Simulate a SysML Model	1570
Create a Requirements Model	1572
Create an Operational Domain Model	1573
Compose System Design	1575
Create Reusable Subsystems	1577
Migrate SysML 1.1 Model to SysML 1.2	1578

Part XIII SOA and XML 1580

XML Schema - XSD	1581
Model XSD	1581
Schema Package.....	1583
Global Element.....	1584
Local Element.....	1585
Global Attribute.....	1586
Local Attribute.....	1587
Attribute Group.....	1588
Complex Type.....	1588
Model Group.....	1590
Group.....	1591
Simple Type.....	1591
Union.....	1592
Enumeration.....	1593
Any.....	1594
Any Attribute.....	1595
Abstract XSD models	1595
Default UML to XSD Mappings.....	1597
Generate XSD	1598
Generate Global Element.....	1599
Import XSD	1600
Global Element and ComplexType.....	1601
UML Profile for XSD	1602
Profile for XML Schema.....	1603
XSD Datatypes Package.....	1610
Web Services - WSDL	1611
Create WSDL 1.1 Model Structure	1611
Model WSDL	1612

WSDL Message.....	1613
WSDL Message Part.....	1614
WSDL Port Type.....	1614
WSDL Port Type Operation.....	1615
WSDL Binding.....	1617
WSDL Binding Operation.....	1618
WSDL Service.....	1619
WSDL Document.....	1620
Generate WSDL	1621
Import WSDL	1622
SoaML	1624
SOMF	1628
Generate MOF	1630
Getting Started	1631
Export MOF to XMI	1632
Part XIV Visual Execution Analyzer	1635
Recording Sequence Diagrams	1637
How it Works	1638
Setup for Recording	1639
Pre-Requisites.....	1639
Configure Recording Detail.....	1639
Control Stack Depth.....	1641
Place Recording Markers	1642
Marker Types	1642
Set Record Markers.....	1644
Recording Activity for a Single Method.....	1645
Recording Activity for a Class	1646
The Breakpoints & Markers Window	1646
Activate and Disable Markers.....	1647
Working with Marker Sets.....	1648
Control the Recording Session	1648
Recorder Toolbar	1648
Start Recording.....	1650
State Transition Recorder	1650
Generating Sequence Diagrams	1651
The Recording History.....	1651
Generate a Diagram.....	1652
Diagram Features.....	1652
Saving Recording.....	1653
Add State Transitions	1653
Setup for Capturing State Changes	1653
The State Machine.....	1655
Recording and Mapping State Changes.....	1657
Profiling Native Applications	1658
System Requirements	1658
Getting Started	1659
Start & Stop the Profiler	1660
Profiler Operation	1661
Setting Options	1661
Save and Load Reports	1662
Save Report in Team Review	1662
Object Workbench	1663
Workbench Setup	1663
How it Works	1664
Workbench Variables	1665

Create Workbench Variables	1665
Invoke Methods	1666
Unit Testing	1669
Set Up Unit Testing	1669
Run Unit Tests	1670
Record Test Results	1670
Testpoint Management	1672
Test Case	1674
Test Cut	1675
Test Set	1675
Test Suite	1676
The Testpoints Window	1676
The Testpoints Window Toolbar	1678
Class Invariant Dialog.....	1678
Operation Constraints Dialog.....	1679
Model Simulation	1681
How It Works	1681
Set Up Simulation	1681
Activate Simulation Script	1682
Run Model Simulation	1683
Using Simulation Breakpoints	1684

Part XV Testing 1687

Model Validation	1688
Configure Model Validation	1689
Rules Reference	1690
Well-Formedness	1691
Element Composition.....	1691
Property Validity.....	1692
OCL Conformance.....	1692
Testing	1694
The Testing Workspace	1695
The Test Details Dialog	1696
Unit Testing	1696
Integration Testing	1698
System Testing	1699
Acceptance Testing	1700
Scenario Testing	1701
Move or Copy Tests Between Categories	1703
Import Scenario as Test	1703
Import Test From Other Elements	1705
Import Responsibility or Constraint as Test	1706
Create Maintenance Item From Test	1707
Show Test Script Compartments	1707
Test Documentation	1708

Part XVI Maintenance 1710

Maintenance Diagram	1711
Example Maintenance Diagram	1712
Maintenance	1713
The Maintenance Workspace	1713
Maintenance Item Properties	1715
Move or Copy Maintenance Items	1716
Create Elements From Maintenance Item	1717
Show Maintenance Script in Diagram	1717

Changes and Defects	1719
Defects (Issues)	1719
Changes	1721
Element Properties	1722
Assign People to Defects or Changes	1722
 Part XVII Reporting	 1724
RTF Documents	1726
Generate RTF Documents	1727
Diagram Options.....	1729
Report Elements From External Packages.....	1730
Exclude Package from Report.....	1731
Generate RTF Documentation Dialog.....	1732
RTF Templates Tab.....	1734
RTF Style Template Editor.....	1736
Select Components for Reporting.....	1737
Tabular Sections.....	1738
Child Sections.....	1740
Constraint and Scenario Sections.....	1741
Add Content.....	1742
RTF Style Template Editor Options.....	1743
Scroll Through Text.....	1744
File and Print Options.....	1745
Cut and Paste Options.....	1746
View Options.....	1747
Image and Object Inserts.....	1749
Character Formatting.....	1750
Paragraph Formatting.....	1751
Tab Support.....	1753
Page Breaks and Repagination.....	1754
Headers and Footers.....	1754
Hyperlinks and Bookmarks.....	1755
Table Commands.....	1756
Sections and Columns.....	1759
Stylesheets and Table of Contents.....	1759
User-Defined Section Numbering.....	1761
Frames and Drawing Objects.....	1763
Search and Replace Commands.....	1764
Import RTF Template.....	1765
Resource Documents.....	1766
Document Options.....	1767
Element Filters.....	1770
Other Filters.....	1771
Project Constants.....	1772
Word Substitution.....	1773
Language Substitution.....	1773
Virtual Documents	1774
Create Master Document.....	1776
Create Model Document.....	1777
Add Packages to Model Document.....	1778
Delete Package in Model Document.....	1779
Document Order.....	1780
Section Numbering in Virtual Documents.....	1782
Generate the Document (RTF).....	1782
Other Documents	1783
Dependency Report.....	1783
Diagrams Only Report.....	1784

Implementation Report.....	1785
Implementation Targets Dialog.....	1785
Testing Report.....	1786
Testing Details Report.....	1786
The Legacy RTF Report Generator	1787
Document a Single Element.....	1788
Set the Main RTF Properties.....	1788
Apply a Filter.....	1789
Exclude Elements.....	1790
RTF Diagram Format.....	1790
Model Include.....	1790
RTF Report Options.....	1790
RTF Report Selections.....	1791
Generate the Report.....	1792
Legacy RTF Style Templates.....	1792
Save as Document.....	1793
Custom Language Settings.....	1794
Use MS Word	1795
Open a Report in Microsoft Word.....	1796
Change Linked Images to Embedded.....	1796
RTF Bookmarks.....	1796
Other Features of Word.....	1798
Add Table of Contents.....	1798
Add Table of Figures.....	1799
Add Headers and Footers.....	1799
Manipulate Tables in Word.....	1799
Refresh Links.....	1801
HTML Reports	1802
Create an HTML Report	1802
The Generate HTML Report Dialog	1804
Web Style Templates	1804
HTML Template Fragments	1806
Master Documents and Model Documents	1813
Generate a Master Document (HTML).....	1814

Part XVIII Automation 1816

Scripting	1817
Scripts Tab	1818
Script Group Properties.....	1819
Console Tab	1820
Enterprise Architect Object Model	1822
Using the Automation Interface	1822
Connect to the Interface.....	1823
Set References In Visual Basic.....	1825
Examples and Tips.....	1826
Call from Enterprise Architect.....	1827
Available Resources.....	1828
Reference	1828
Interface Overview Package.....	1829
App Object.....	1830
Enumerations.....	1831
ConstLayoutStyles.....	1832
CreateBaselineFlag.....	1832
CreateModelType.....	1832
DocumentBreak.....	1833
DocumentType.....	1833
EA EditionTypes.....	1833

EnumRelationSetType.....	1834
ExportPackageXMIFlag.....	1834
MDGMenus.....	1834
MessageFlag.....	1835
ObjectType.....	1835
PropType.....	1836
ReloadType.....	1837
ScenarioDiagramType.....	1837
ScenarioStepType.....	1838
ScenarioTestType.....	1838
XMIType.....	1838
Repository Package.....	1838
Author Class.....	1839
Client Class.....	1840
Collection Class.....	1841
Datatype Class.....	1843
EventProperties Class.....	1844
EventProperty Class.....	1844
ModelWatcher Class.....	1845
Package Class.....	1845
ProjectIssues Class.....	1851
ProjectResource Class.....	1852
ProjectRole Class.....	1853
PropertyType Class.....	1853
Reference Class.....	1854
Repository Class.....	1855
Stereotype Class.....	1871
Task Class.....	1872
Term Class.....	1873
Element Package.....	1873
Constraint Class.....	1876
Effort Class.....	1876
Element Class.....	1877
File Class.....	1887
Issue (Maintenance) Class.....	1888
Metric Class.....	1889
Requirement Class.....	1890
Resource Class.....	1891
Risk Class.....	1891
Scenario Class.....	1892
ScenarioExtension Class.....	1893
ScenarioStep Class.....	1894
TaggedValue Class.....	1895
Test Class.....	1896
Element Features Package.....	1897
Attribute Class.....	1898
AttributeConstraint Class.....	1900
AttributeTag Class.....	1901
CustomProperties Collection.....	1902
EmbeddedElements Collection.....	1902
Method Class.....	1903
MethodConstraint Class.....	1905
MethodTag Class.....	1906
Parameter Class.....	1906
ParamTag Class.....	1908
Partitions Collection.....	1908
Properties Class.....	1909
Transitions Collection.....	1910

Connector Package.....	1910
Connector Class	1911
ConnectorConstraint Class.....	1915
ConnectorEnd Class	1916
ConnectorTag Class	1918
RoleTag Class.....	1919
Diagram Package.....	1920
Diagram Class.....	1921
DiagramLinks Class.....	1924
DiagramObject Class.....	1925
SwimlaneDef Class.....	1927
Swimlanes Class	1927
Swimlane Class	1928
Project Interface Package.....	1929
Project Class.....	1929
Document Generator Interface Package.....	1947
DocumentGenerator Class.....	1947
Mail Interface Package.....	1950
MailInterface Class.....	1950
Code Samples	1952
Open the Repository.....	1953
Iterate Through a .EAP File.....	1953
Add and Manage Packages.....	1954
Add and Manage Elements	1955
Add a Connector.....	1956
Add and Manage Diagrams.....	1957
Add and Delete Features.....	1957
Element Extras	1958
Repository Extras	1961
Stereotypes	1963
Work With Attributes.....	1964
Work With Methods.....	1965
Enterprise Architect Add-In Model	1967
Add-In Tasks	1968
Create Add-Ins.....	1968
Define Menu Items.....	1969
Deploy Add-Ins	1970
Tricks and Traps.....	1972
The Add-In Manager	1974
Add-In Search	1974
XML Format (Search Data).....	1975
Add-In Events	1976
EA_Connect.....	1977
EA_Disconnect.....	1977
EA_GetMenuItems.....	1978
EA_GetMenuState.....	1979
EA_MenuClick.....	1980
EA_OnOutputItemClicked.....	1981
EA_OnOutputItemDoubleClicked.....	1982
EA_Show Help.....	1983
Broadcast Events	1983
Add-In License Management Events.....	1984
EA_AddinLicenseValidate.....	1985
EA_AddinLicenseGetDescription.....	1986
Compartment Events	1986
EA_QueryAvailableCompartments.....	1986
EA_GetCompartmentData	1987
Context Item Events.....	1989

EA_OnContextItemChanged.....	1990
EA_OnContextItemDoubleClicked.....	1991
EA_OnNotifyContextItemModified.....	1992
EA_FileClose.....	1993
EA_FileNew.....	1993
EA_FileOpen.....	1994
EA_OnPostCloseDiagram.....	1995
EA_OnPostInitialized.....	1995
EA_OnPostOpenDiagram.....	1996
EA_OnPostTransform.....	1997
EA_OnPreExitInstance.....	1997
EA_OnRetrieveModelTemplate.....	1998
Model Validation Broadcasts.....	1999
EA_OnInitializeUserRules.....	2000
EA_OnStartValidation.....	2000
EA_OnEndValidation.....	2001
EA_OnRunElementRule.....	2001
EA_OnRunPackageRule.....	2002
EA_OnRunDiagramRule.....	2002
EA_OnRunConnectorRule.....	2003
EA_OnRunAttributeRule.....	2004
EA_OnRunMethodRule.....	2005
EA_OnRunParameterRule.....	2005
Model Validation Example.....	2006
Post-New Events.....	2010
EA_OnPostNew Element.....	2011
EA_OnPostNew Connector.....	2011
EA_OnPostNew Diagram.....	2012
EA_OnPostNew DiagramObject.....	2013
EA_OnPostNew Attribute.....	2014
EA_OnPostNew Method.....	2015
EA_OnPostNew Package.....	2015
EA_OnPostNew Glossary Term.....	2016
Pre-Deletion Events.....	2017
EA_OnPreDeleteElement.....	2017
EA_OnPreDeleteAttribute.....	2018
EA_OnPreDeleteMethod.....	2019
EA_OnPreDeleteConnector.....	2020
EA_OnPreDeleteDiagram.....	2020
EA_OnPreDeleteDiagramObject.....	2021
EA_OnPreDeletePackage.....	2022
EA_OnPreDeleteGlossary Term.....	2023
Pre-New Events.....	2023
EA_OnPreNew Element.....	2024
EA_OnPreNew Connector.....	2025
EA_OnPreNew Diagram.....	2026
EA_OnPreNew DiagramObject.....	2026
EA_OnPreNew Attribute.....	2027
EA_OnPreNew Method.....	2028
EA_OnPreNew Package.....	2029
EA_OnPreNew Glossary Term.....	2030
Tagged Value Broadcasts.....	2031
EA_OnAttributeTagEdit.....	2031
EA_OnConnectorTagEdit.....	2032
EA_OnElementTagEdit.....	2033
EA_OnOperationTagEdit.....	2034
Technology Events.....	2034
EA_OnInitializeTechnologies.....	2035

EA_OnPreActivateTechnology	2036
EA_OnPostActivateTechnology.....	2037
EA_OnPreDeleteTechnology.....	2037
EA_OnDeleteTechnology.....	2038
EA_OnImportTechnology	2039
Custom Views	2040
Create a Custom View	2041
Custom Docked Window	2042
MDG Add-Ins	2042
MDG Events.....	2042
MDGBuild Project.....	2043
MDGConnect.....	2044
MDGDisconnect.....	2045
MDGGetConnectedPackages.....	2046
MDGGetProperty.....	2046
MDGMerge.....	2047
MDGNew Class.....	2049
MDGPostGenerate.....	2050
MDGPostMerge.....	2051
MDGPreGenerate.....	2051
MDGPreMerge.....	2052
MDGPreReverse.....	2053
MDGRunExe.....	2054
MDGView	2054

Part XIX Glossary 2057

A	2058
B	2060
C	2061
D	2064
E	2066
F	2068
G	2069
H	2070
I	2071
J	2073
L	2074
M	2075
N	2077
O	2078
P	2079
Q	2082
R	2083
S	2085
T	2088
U	2090
V	2092

Part XX License Management 2094

Finding Your License Information	2095
Adding License Keys	2096
Keystore Troubleshooting	2098
Upgrade an Existing License	2099
Register Add-In	2101
Index	2102

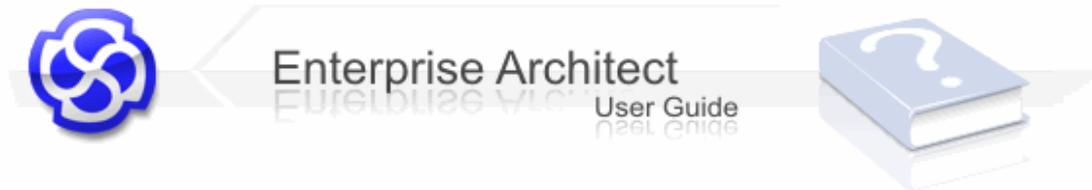
Foreword

This user guide provides an introduction to the features contained in Enterprise Architect - a UML modeling tool for developing and building software systems with UML.

Part



1 Introduction



Welcome to Sparx Systems **Enterprise Architect**, a UML based modeling tool for designing and constructing software systems, business modeling, systems modeling and generalized modeling purposes such as visualizing existing systems and processes.

The *Enterprise Architect User Guide* provides reference materials, guidance and tutorials for using the features and facilities of Enterprise Architect in your modeling and in related work.

General Topics:

Topic	Link
Enterprise Architect is very flexible and has many features; the Overview provides a summary of what Enterprise Architect can do and what you can use it for	What You Can Do ^[5] Key Benefits ^[7] Key Features ^[10]
<p>If you are new to modeling and UML as well as Enterprise Architect, or otherwise want a rapid review of the process of modeling with Enterprise Architect, the <i>User Guide</i> provides a tutorial to help you get working quickly</p> <p>The tutorial is not just a theoretical description - the first things you do are start Enterprise Architect and immediately create a model project</p>	Getting Started ^[34] A Quickstart Tutorial ^[37]
When working through the Quickstart tutorial you should see many links to more extensive descriptions of features, functions, tasks and procedures ; you can follow any of these immediately if you require more information	User Interface Guide ^[69]
You should also read the Sparx Systems Formal Statements , including the Copyright Notice and our End User Licensing Agreement.	Formal Statements ^[19]
The <i>User Guide</i> provides a glossary of terms that you can refer to for definitions of various terms and concepts used in this guide	Glossary ^[2057]
<p>Feedback - Sparx Systems likes to stay in touch with what Enterprise Architect users require in order to accomplish their tasks efficiently and effectively.</p> <p>We value any suggestions, feedback and comments you might have regarding this product, documentation or install process</p> <p>You can provide your feedback:</p> <ul style="list-style-type: none"> • Using a fault (bug) report or feature request, or • By email 	www.sparxsystems.com/bug_report.htm www.sparxsystems.com/feature_request.htm support@sparxsystems.com
If you are evaluating the free trial version of Enterprise Architect and have not yet purchased an edition, but would like to, you can refer to the information we provide to help you make your purchase	Order Enterprise Architect ^[26]

Specific Topics:

Topic	Link
Setting up , configuring, managing and maintaining modeling projects , including storage mechanisms, team development environments, managing changes and monitoring progress	Projects and Teams <small>[135]</small>
Creating and maintaining model structures in Enterprise Architect, including how to work with the model structure and components, their properties, and the reference data with which you populate the model	Modeling Basics <small>[515]</small>
Understanding the Enterprise Architect implementation of UML concepts and specifications , how those concepts are extended in Enterprise Architect's support of other modeling languages, and how you can extend the concepts yourself by creating your own modeling languages	Define a Modeling Language <small>[1038]</small>
Navigating through your model, searching for information you require on the model, selecting aspects of the model to extract and review, and tracing developmental relationships in both the structure and the development timeframe	Navigate, Search & Trace <small>[440]</small>
Transforming model elements and fragments from one domain to another, using standard and customized Model Driven Architecture transformations	Model Transformation <small>[1301]</small>
Generating code from the model (forward engineering), generating model structures from imported code (reverse engineering) synchronizing code and model, controlling how these operations take place and controlling what output results from the processes	Software Engineering <small>[1388]</small>
Performing visual analysis of executing code - debugging, recording the stack trace and generating Sequence diagrams from the trace	Visual Execution Analyzer <small>[1635]</small>
Developing models for specific modeling domains , such a business process analysis or systems engineering	Analysis and Business Modeling <small>[1190]</small> Database Engineering <small>[1345]</small> Systems Engineering <small>[1548]</small> SOA and XML Engineering <small>[1580]</small>
Ensuring quality control of your models and code through model validation and running your own test scripts	Testing <small>[1687]</small>
Managing the maintenance of model elements and changes and issues across the project	Maintenance <small>[1710]</small>
Documenting your model in either RTF or HTML output format	Reporting <small>[1724]</small>
Using the Technology Developer's tools , which enable you to access Enterprise Architect facilities through your own interface, in your own implementation	Automation <small>[1816]</small>
Managing your private or shared Enterprise Architect product license keys	License Management <small>[2094]</small>

1.1 Overview

Enterprise Architect is a powerful CASE tool for specifying, documenting and building software projects. Using Enterprise Architect's support for UML and its related standards, you can model new complex software and business systems, or visualize and maintain existing systems.

Topics:

Topic	Link
<p>Enterprise Architect is a comprehensive UML analysis and design tool, covering all aspects of the software development cycle from requirements gathering through analysis, model design, testing, change control and maintenance to implementation, with full traceability</p> <p>Enterprise Architect combines the power of the latest UML 2.3 specification with a high performance, intuitive interface, to bring advanced modeling to the whole development team</p> <p>It is a multi-user, visual tool with a great feature set, helping analysts, testers, project managers, quality control staff and deployment staff around the world to build and document robust, maintainable systems and processes</p>	<p>Key Benefits of Enterprise Architect ^[7]</p> <p>Enterprise Architect Key Features ^[10]</p>
<p>With over 250,000 licenses sold, Enterprise Architect has proven highly popular, being used by thousands of companies world-wide</p> <p>From large, well-known, multi-national organizations to smaller independent companies and consultants, Enterprise Architect has become the UML modeling tool of choice for developers, consultants and analysts in over 130 countries</p> <p>Sparx Systems software is used in the development of many kinds of application and system in a wide range of industries including:</p> <ul style="list-style-type: none"> • aerospace • banking • web development • engineering • finance • medicine • military • research • academia • transport • retail • utilities (such as gas and electricity) and • electrical engineering <p>It is also used effectively for UML and enterprise architecture training in many prominent colleges, training companies and universities around the world</p>	

1.1.1 What You Can Do

This topic introduces the fundamental processes that Enterprise Architect supports.

Topic	Detail	See also
Modeling	Enterprise Architect is a comprehensive model analysis and design tool	Modeling Fundamentals ^[5]

	<p>To create models with Enterprise Architect, you therefore should become familiar with:</p> <ul style="list-style-type: none"> • How Enterprise Architect implements the UML standards and • How you apply UML in Enterprise Architect to develop your models 	
Managing Models	<p>To manage the models in your projects, you both</p> <ul style="list-style-type: none"> • Protect and manage the model data itself, and • Communicate information on the data in the form of RTF and HTML documentation and reports 	Projects and Teams ^[135] Reporting ^[1724]
Code Engineering	<p>In Enterprise Architect, UML modeling both depends on and supports code engineering - you generate and update code from a model, and you create and update models from code</p> <p>In this broad sense, Enterprise Architect enables you to:</p> <ul style="list-style-type: none"> • Forward engineer, reverse engineer, round-trip and synchronize code in a range of programming languages • Debug and profile code • Model and generate code for XML Technologies • Perform database modeling and design for a range of database management systems • Convert model components from one domain to another using Model Driven Architecture (MDA) Transformations. 	Overview Of Development ^[1389]
Managing Projects	<p>Enterprise Architect provides strong support for Project Management, particularly in the following areas:</p> <ul style="list-style-type: none"> • Project estimation - working out how much time and effort is required to build and deploy a solution, using the Use Case metrics facility and carefully-calibrated metrics • Defining, assigning and managing resources • Communicating Project Management discussions and decisions to the project • Monitoring and managing problems, changes, issues and tasks that affect both individual elements and the project as a whole • Managing the development, execution and results of testing, from Integration through to User Acceptance, and • Maintaining a project glossary of terms, procedures and policies applied to the project <p>The scope of your project management might include upgrades to Enterprise Architect and installation of related technologies</p>	Projects and Teams ^[135] Testing ^[1687] Maintenance ^[1710] License Management ^[2094]
Extending Enterprise Architect Facilities	<p>Experienced Technology Developers can develop customized additions to the functionality already present within Enterprise Architect. These additions include:</p> <ul style="list-style-type: none"> • UML Profiles and Stereotypes • UML Patterns • Code Templates 	Build Your Own Modeling Language ^[1038] Automation ^[1816]

	<ul style="list-style-type: none"> • Tagged Value Types • MDG Technologies and • Enterprise Architect Add-Ins. <p>By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.</p>	
--	--	--

1.1.2 Key Benefits

Enterprise Architect is a powerful tool for specifying, documenting and building your software and business process projects. Using Enterprise Architect's **support for UML** and its related standards, you can model new complex software and business systems, or visualize and maintain existing systems.

Topic	Detail	See also
Design and Build Diverse Systems Using UML	<p>UML 2.3 is an open standard that provides a rich language for describing, documenting and designing software, business and IT systems in general</p> <p>Enterprise Architect enables you to leverage the full expressive power of UML 2.3 to model, design and build diverse systems in an open and well understood manner</p> <p>You can generate code, database structures, documentation and metrics; transform models; or specify behavior and structure as the basis for contractual agreements</p>	<p>Standard UML Models ¹⁷⁹³</p> <p>Modeling Basics ⁵¹⁵</p>
Model and Manage Complexity	<p>Enterprise Architect helps individuals, groups and large organizations model and manage complex information.</p> <p>Often this relates to software development and IT systems design and deployment, but it can also relate to business analysis and business process modeling</p> <p>Enterprise Architect integrates and connects a wide range of structural and behavioral information, helping to build a coherent and verifiable architectural model, either what-is or what-will-be</p> <p>Tools to manage version control, track and compare differences, audit changes and enforce security, help control project development and enforce compliance to standards</p>	<p>Configure User Security ¹⁹⁴</p> <p>Change Management ²⁴</p>
Structured Use Case Scenarios	<p>Enterprise Architect's Structured Scenario editor enables you to develop structured Use Case Scenarios, to capture vital analysis information in the form of natural language descriptions</p> <p>The editor helps you use this information to drive downstream development and maximize traceability across the development life-cycle</p> <p>The editor also helps you to dynamically link scenario steps to associated model elements, such as domain elements, business rules and glossary terms</p> <p>From structured scenarios, you can automatically generate test case descriptions, and Activity and other UML behavioral diagrams</p> <p>You can even reverse engineer existing process diagrams</p>	<p>Scenarios ⁶⁶⁷</p>

Topic	Detail	See also
	into structured, textual specifications to produce documentation deliverables	
Share Models	<p>Enterprise Architect enables you to share complete models or specific aspects of a model between members of a team, including (through the 'Lite', read-only edition) stakeholders who can study a model but not change or manage it</p> <p>You can make your project .EAP file available on a shared network drive, or replicate the .EAP file for complex distributed development</p> <p>Alternatively, you can develop the project in one of several shared DBMS repositories</p> <p>You can import and export data as XMI files to distribute and update frameworks and other package-based model structures; you control changes through the version control repository</p> <p>Enterprise Architect provides a data transfer wizard that enables you to upsize or downsize the complete model for maximum flexibility, and it enables you to export and import reference data so that you do not have to recreate it for related projects</p>	<p>The Read-only 'Lite' Edition ^[18]</p> <p>Project Sharing ^[185]</p> <p>Server Based Repositories ^[148]</p> <p>XMI Import and Export ^[319]</p> <p>Version Control ^[242]</p> <p>Perform a Project Data Transfer ^[343]</p> <p>Sharing Reference Data ^[236]</p>
Model, Manage and Trace Requirements	<p>Enterprise Architect enables you to capture requirements and use full traceability from base requirements to design, build, deployment and beyond</p> <p>You can use impact analysis to trace from proposed changes to original requirements, and build the 'right' system</p>	<p>Requirements ^[1152]</p> <p>Trace: Tracking Dependencies ^[493]</p>
Develop Personal Views and Extracts of the Model	<p>Enterprise Architect enables you to develop any number of different views of your model, or parts of it, either for your personal use or for the use of your team</p> <p>These Model Views are generated by reports, so they can be set up to always show the current status of the selected view</p> <p>The facility also enables you to create Favorites folders of hyperlinks to frequently-used data structures</p>	<p>Model Views ^[464]</p>
Track and Trace Model Structures	<p>In even a small model, it can be difficult to locate specific packages, diagrams or elements, even if you apply a rigorous naming and structure policy</p> <p>Enterprise Architect has a wealth of facilities that enable you to locate structures quickly and easily, through the Model Search, Package Browser, Diagram List, Auditing facility, Traceability window, Relationship Matrix and reports</p> <p>The Element menu, Diagram menu and Project Browser context menus also enable you to locate elements in diagrams and in the Project Browser, and you can store hyperlinks to important or commonly-used elements and diagrams in the Model Views</p> <p>Finally, having located one element you can import any related elements into a diagram in a single operation</p>	<p>Navigate, Search & Trace ^[440]</p> <p>Auditing ^[298]</p> <p>Other Documents ^[1783]</p> <p>Main Menu ^[73]</p> <p>Project Browser Context Menus ^[443]</p> <p>Insert Related Elements ^[652]</p>

Topic	Detail	See also
Generate Documentation	<p>Enterprise Architect provides powerful document generation and reporting tools with a full WYSIWYG template editor for RTF or HTML output</p> <p>You can generate complex and detailed reports from Enterprise Architect with the information you require in the format your company or client demands</p>	Reporting ^[1724]
Generate and Reverse Engineer Source Code	<p>Enterprise Architect supports generation and reverse engineering of source code for many popular languages</p> <p>With a built in 'syntax highlighting' source code editor, Enterprise Architect enables you to quickly navigate and explore your model source code in the same environment</p> <p>Code generation templates enable you to customize the generated source code to your company specifications</p>	Generate Source Code ^[1490] Importing Source Code ^[1508] The Source code Viewer ^[1408] Code Templates ^[1482]
Visualize, Inspect and Understand Complex Software	<p>Software is complex and often hard to understand; you can use Enterprise Architect to reverse engineer code in a wide range of software development languages and database repository schema, to understand static structure</p> <p>To complete the picture, use the unique built-in profiling and debugging tools to capture and visualize executing software at run-time</p> <p>Create run-time instances of model elements and invoke methods using the built in Object Workbench</p> <p>You can also bring in complete frameworks from source code or Java .jar files - or even .Net binary assemblies!</p> <p>By importing frameworks and library code, you can maximize re-use and understanding of your existing investment</p>	Importing Source Code ^[1508] Execution Analysis ^[1635] Object Workbench ^[1663] Import Source Code ^[1510] Import Binary Module ^[1513]
Perform MDA Transformations	<p>Model Driven Architecture (MDA) is an open standard designed to facilitate rapid application development in a platform independent manner</p> <p>Models can be built at a high level of abstraction and, using MDA based tools, transformed into models and code targeting a specific platform or domain</p> <p>Enterprise Architect supports advanced MDA transformations using easily edited and developed transformation templates</p> <p>With built-in transformations for DDL, C#, Java, EJB and XSD, you can quickly develop complex solutions from simple platform independent models (PIMs) targeted at platform specific models (PSMs); one PIM can be used to generate and synchronize multiple PSMs, providing a significant productivity boost</p>	Model Transformation ^[1307] Built-in Transformations ^[1307]
SOA (Service Oriented Architecture) Support	<p>Enterprise Architect enables you to rapidly model and forward- and reverse-engineer two key W3C XML technologies:</p> <ul style="list-style-type: none"> • XML Schema (XSD) and • Web Service Definition Language (WSDL) <p>XSD and WSDL support is critical for the development of a complete Service Oriented Architecture (SOA), and the</p>	XML Schema - XSD ^[1581] Web Services - WSDL ^[1611]

Topic	Detail	See also
	coupling of UML 2.3 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization	
Systems Engineering support	Integrating many high-end features for Systems Engineers, the Ultimate and Systems Engineering editions of Enterprise Architect provide built-in support for: <ul style="list-style-type: none"> • SysML • Parametric model simulation • Executable code generation • Model to code transformations for Hardware Description Languages and Ada 2005 	Editions Available ^[14] SysML ^[1552] Behavior ^[706] State Machine Modeling For HDLs ^[1502] Ada 2005 ^[1459]
Model Databases	Enterprise Architect enables you to reverse engineer from many popular DBMS systems You can model database tables, columns, keys, foreign keys and complex relationships using UML and an inbuilt data modeling profile, and forward generate DDL scripts to create target database structures	Supported Databases Data Models ^[1276]
Customize Enterprise Architect	Enterprise Architect also includes facilities that enable experienced tool developers to customize and extend Enterprise Architect to suit the specific requirements of their organization with, for example, in-house UML Profiles, Add-Ins and Code Templates The very detailed Automation Interface gives you access to most element features, major functions such as XML import/export, and attached information; most properties are fully writable from the automation client The Automation Interface provides great support for plug-ins, with the ability to embed automation client windows in the main diagram view The Interface is accessible from any automation-aware client language, such as VB, C#, C++ and Delphi	Developing Profiles ^[1039] Enterprise Architect Add-In Model ^[1967] Code Template Framework ^[1114] Using the Automation Interface ^[1822]
Link Enterprise Architect to IDEs	Using Sparx Systems Model Driven Generation (MDG) <i>Link</i> plug-ins, you can develop source code in your preferred Integrated Development Environment such as Visual Studio .NET or Eclipse, while you use Enterprise Architect to locate the source code for Classes, attributes and operations, and to model, navigate, track, reverse engineer, build and run your project The MDG Integration products for Eclipse and Visual Studio 2008 provide an even closer, seamless integration of Enterprise Architect and UML 2.3 with your IDE, bringing the functionality required of a fully fledged modeling platform right inside the IDE	Visual Studio .NET Eclipse (Link) Eclipse (Integration) Visual Studio 2008

1.1.3 Key Features

Enterprise Architect is renowned for its rich feature set. Some of the key features are highlighted in the following list:

- Model complex information, software and hardware systems using UML-compliant notation (comprehensive **UML 2.3** support for all 14 UML diagrams)
- Extended modeling for **Requirements, User Interface Design, Mind Mapping, Data Modeling, SysML,**

SPEM, BPMN 1.1 and more

- Generate **BPEL** scripts automatically from **Business Process** models
- Built-in **Requirements Management** enables you to specify, trace and verify requirements directly against the design, right through to the deployed solution
- Comprehensive and flexible MS Word-compatible **HTML and RTF report options**
- Leverage industry-standard **Enterprise Architecture** frameworks (**Zachman, TOGAF, DoDAF-MODAF**)
- Support in **forward and reverse code engineering** for many software and hardware languages 'out of the box': ActionScript 3.0, Java, C#, C++, VB.Net, Delphi, Visual Basic, Python, PHP, Verilog, VHDL and SystemC
- Ability to perform **database modeling**, to **reverse engineer** from a range of DBMSs via ODBC, and to **forward generate DDL scripts** to create database structures
- Connect to **shared database repositories** using MS SQL Server, MySQL, Oracle and more
- **Manage, track and control change** using **baseline** model merge and **auditing** capabilities
- **Centralize enterprise-wide documentation** of processes and information systems
- **Model dependencies** between elements, system dynamics and state
- **Model class hierarchies**, deployment, components and implementation details
- **Record project issues, tasks** and system glossary
- **Assign resources** to model elements and **track effort expended** against required effort
- **Testing support** for test cases, JUnit and NUnit
- Integrated **Debug Workbench** for visualizing executable Java and .Net applications, instantiating runtime model objects and generating Sequence diagrams from a stack trace
- **Migrate changes** across a distributed environment using **Controlled XMI Packages**
- Manage **Version control** through XMI using **SCC CVS** and **Subversion** configurations
- Inbuilt user and group **security** and access control management
- **Distributed development** through shareable files, use of **shared repositories** in a range of major Database Management Systems, file replication, data transfer, and import and export of reference data
- **Share models** using the latest **XMI 2.1** format
- **Import models** in XMI format from other tools
- Built-in Model Driven Architecture (**MDA**) **Transformations**, and facilities to import or create others
- Facilities to **import database schema, XSD and WSDL source, .NET and Java binaries**
- Use **UML Profiles** to create custom extensions for domain-specific modeling
- Save and load complete diagrams as **UML Patterns**
- Create and share dynamic views of model elements and diagram sets using **Model Views**
- Analyze and trace relationships between elements using the tabular **Relationship Matrix**
- Generate **executable business logic** from **rule tasks** and trace to natural language **business rules**
- Transform **behavioral models** into executable source code for software and **hardware description languages** (HDLs) such as Verilog, VHDL, and SystemC
- **Simulate SysML parametric models**
- Script and automate common tasks using a detailed **Automation Interface** and **Model Scripts**
- A range of internal and external commercial MDG Add-Ins to integrate the facilities of Enterprise Architect with IDEs and other technologies, and templates to write your own
- **Read-only Viewer** enables stakeholders to view but not change milestone deliverables
- **Price:** Enterprise Architect is priced to outfit the entire team, making collaboration and team development a real possibility
- **Speed:** Enterprise Architect is quick to load and a spectacularly fast performer, even with large models
- **Scalability:** Enterprise Architect supports single users and the development of small models, or many concurrent users developing extremely large models, with equal ease
- **Usability:** many of our users agree, Enterprise Architect gets you started and productive quickly, with a rich user interface and the ability to create **templates, model views** and 'favorites' collections of commonly-used elements and diagrams.

For a complete list of the new features of the latest version of Enterprise Architect, click on the **Help | Read Me** menu option.

Enterprise Architect is available in six editions: **Ultimate**, **Business and Software Engineering**, **Systems Engineering**, **Corporate**, **Professional**, and **Desktop**, each of which offers a different range of features.

Learn More:

- [Editions Available](#)^[14]
- [Extensions - MDG Technologies](#)^[1036]

1.2 Enterprise Architect Editions



Enterprise Architect is available in a number of different editions, each tailored to support a particular business case.

Topic	Detail	See also
Before Purchase	Test the product in a number of configurations in the Trial Version	The Trial Version ^[13]
Using the Trial	Explore the specific features of the six work environment editions available	Editions Available ^[14]
Reviews and other Read-only tasks	Use the free, read-only or 'Lite' edition	The Read-only 'Lite' Edition ^[18]

1.2.1 The Trial Version

The fully functional 30 day trial version of Enterprise Architect is available free of charge on the Sparx Systems website www.sparxsystems.com/bin/easetup.exe. The trial version is identical to the registered edition with the exception that all diagrams are output to files with an embedded watermark.

Topic	Detail	See also
Try Out Editions	<p>The trial version enables you to explore and evaluate Enterprise Architect, checking the edition you are interested in and trying out other editions for comparison</p> <p>When you start up the Enterprise Architect trial version, the Select Trial Version prompt displays; select the mode to trial</p> <p>If necessary, you can close down Enterprise Architect and restart it in another mode for comparison</p> <p>The prompt also directs you to useful information such as a walkthrough of the Enterprise Architect facilities, and enables you to use one of several workspace layouts</p>	Workspace Layouts ^[15]
Trial Period:	<p>As you are evaluating the Enterprise Architect trial version, note that the software operates for a limited period, and denies access after the trial period has elapsed</p> <p>To continue using Enterprise Architect when the trial period expires, you can either:</p> <ul style="list-style-type: none"> • Apply to extend the trial period, or • Purchase and register a full license; on purchase of a suitable license or licenses, the registered version is made available for download <p>The latest information on pricing and purchasing is available on</p>	Order Enterprise Architect ^[26] Installation ^[27] Register a Full License ^[29] sales@sparxsystems.com

Topic	Detail	See also
	<p>the Sparx Systems website; for more information, contact Sparx Systems by email</p> <p>When you order and pay for an edition of Enterprise Architect, you receive installation instructions and the location of the executable files for download</p>	
Extend Trial Period	<p>If you are testing the trial version and require more than 30 days to evaluate it, you can apply to Sparx Systems Sales for an extension of the trial period; Sparx Systems Sales send you an extension key by email</p> <p>The extension can be an additional 30 to 150 days; if you do not request a long enough extension in the first place, you can submit further requests</p> <p>The trial period must expire before you can enter the extension key.</p>	sales@sparxsystems.com

How to:

To extend the trial period, after receipt of your extension key, follow the steps below:

Step	Action	See also
1	<p>Open the Enterprise Architect trial version</p> <p>The Evaluation Version of Enterprise Architect dialog displays; once the trial period has expired, you cannot proceed beyond this dialog without extending the trial period</p>	
2	<p>Press (Ctrl) whilst you click on the Continue Trial button</p> <p>The Upgrade Key dialog displays</p>	
3	<p>In the Upgrade Key field, type or copy-and-paste the extension key you received from Sparx Systems Sales</p>	
4	<p>Click on the OK button</p> <p>Enterprise Architect confirms that your trial period has been extended</p> <p>Your trial period is extended by the period of days stated in the email from Sparx Systems Sales; you can now restart and use the Trial version of Enterprise Architect</p>	

1.2.2 Editions Available

Enterprise Architect is available in six editions: the **Ultimate**, **Business and Software Engineering**, and **Systems Engineering 'suite'** editions, and the **Corporate**, **Professional** and **Desktop** editions.

Topic	Detail	See also
Comparison	<p>The functionality for each edition is described below; the features of and differences between the editions are listed in the table provided on the Sparx Systems website</p> <p>To help you understand the differences between these editions</p>	<p>Sparx Systems website</p> <p>The Trial Version</p>

Topic	Detail	See also
	and the advantages and limitations of each, you can open and use the Trial version of Enterprise Architect to mimic any of the editions	
<p>Ultimate Edition</p> 	<p>The Ultimate edition is designed for power users and those working across multiple domains, providing deep support for Business, for Software Engineering and for Systems Development seamlessly integrated into a single development environment</p> <p>It enables you to drill down to the lowest levels of systems design and construction, with SysML and executable code generation for standard and hardware description languages</p> <p>Business users can leverage BPEL, the Rules Composer and executable UML, in addition to all the advanced features of the other editions of Enterprise Architect</p> <p>Software developers can integrate their Eclipse and Visual Studio projects with their UML models and leverage the advanced executable code generators to target different domains</p> <p>The Ultimate edition enables end to end traceability throughout a global vision of your enterprise - unifying strategy, business process, interfaces, software, rules, data and fine grained systems; powerful tools, domain-specific technologies, frameworks, integration platforms and a consistent, scalable, and robust interface work in unison to help you deliver on the promise of Model Driven Development</p> <p>The Ultimate edition incorporates a number of MDG Technologies and Add-Ins; the Ultimate edition and MDG Technologies are all available in either Fixed License or Floating License form</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently</p> <p>The Ultimate edition provides:</p> <ul style="list-style-type: none"> • Executable Code Generation - support for generating functional source code for State Machines, Interactions and Activities in C, C++, C#, Java and VB.NET • Full round trip support for Hardware Description Languages (Verilog, VHDL and SystemC) including support for generating State Machine code • SysML Simulation Support - including support for simulating SysML 1.2 constraint models with results graphing capabilities • BPEL Generation – transform BPMN 1.1 and BPMN 2.0 Business process models down to BPEL 1.1 and BPEL 2.0 code respectively • Business Rules – trace from abstract business rules down to automatically generated behavioral code 	<p>MDG Technologies </p>
<p>Business and Software Engineering Edition</p>	<p>The Business and Software Engineering edition is aimed at software development professionals, business modelers, architects, requirements experts, project managers and others involved in the design and construction of quality software and business services</p> <p>It combines powerful new features such as executable code generation from UML models, BPEL, advanced scripting and a</p>	<p>MDG Technologies </p>

Topic	Detail	See also
	<p>multi-purpose Rules Composer targeting executable code from Business Domain models; it also bundles licenses for integration products and frameworks such as DoDAF-MODAF, TOGAF and Zachman, to provide advanced model-driven construction tools to tightly bind your code development in Eclipse or Visual Studio</p> <p>The Business and Software Engineering edition incorporates a number of MDG Technologies and Add-Ins; the Business and Software Engineering edition and MDG Technologies are all available in either Fixed License or Floating License form</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently</p> <p>The Business and Software Engineering edition provides:</p> <ul style="list-style-type: none"> • Generation of Behavioral Code from State, Sequence and Activity models, supporting standard programming languages such as Java and .NET • Advanced math functions within the scripting engine • BPEL Generation from BPMN 1.1 and BPMN 2.0 models - including validation and WSDL support • A Business Rules Composer that enables you to build Business Domain models and generate code to implement complex business rules in standard programming languages 	
<p>Systems Engineering Edition</p> 	<p>The Systems Engineering edition is designed for systems and software development professionals working on real-time, embedded and systems solutions</p> <p>It combines new features such as executable code generation from UML models (including support for hardware languages such as Verilog and VHDL), Ada, SysML 1.2, executable SysML Parametric diagrams and advanced scripting</p> <p>It also bundles licenses for DoDAF-MODAF, SysML, DDS, TcSE and integration products to provide powerful model-driven construction tools for the Systems Engineering domain to tightly bind your code development in Eclipse or Visual Studio with the UML/SysML models developed in Enterprise Architect</p> <p>The Systems Engineering edition incorporates a number of MDG Technologies and Add-Ins; the Systems Engineering edition and MDG Technologies are all available in either Fixed License or Floating License form</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently</p> <p>This edition provides:</p> <ul style="list-style-type: none"> • Executable Code Generation - support for generating functional source code for State Machines, Interactions and Activities in C, C++, C#, Java and VB.NET • Full round trip support for Hardware Description Languages, including Verilog, VHDL and SystemC, with support for generating State Machine code 	<p>MDG Technologies </p>

Topic	Detail	See also
	<ul style="list-style-type: none"> • SysML Simulation Support - Includes support for simulating SysML 1.2 constraint models with results graphing capabilities 	
<p>Corporate Edition</p> 	<p>Aimed at larger development teams, the Corporate edition enables you to connect to the following DBMS back ends as the shared repository:</p> <ul style="list-style-type: none"> • MySQL • SQL Server • PostgreSQL, • Sybase Adaptive Server Anywhere • Access 2007 and • Oracle 9i, 10g or 11g <p>This provides additional scalability and improved concurrency over the shared .EAP file approach to model sharing</p> <p>User security, user logins, user groups and user level locking of elements, user/group based security (with locking at diagram and element levels) are also supported</p> <p>Security comes in two modes: in the first mode, all elements are considered 'writeable' until explicitly locked by a user or group; in the second mode, all elements are considered locked until checked out with a user lock</p> <p>The Corporate edition forms the base for the three extended editions described above; like those editions, it is available in either Fixed License or Floating License form</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently</p>	
<p>Professional Edition</p> 	<p>Aimed at work groups and developers, the Professional edition supports shared projects through replication and shared network files</p> <p>This edition has an ActiveX interface for interrogating Enterprise Architect projects and extracting information in XML format</p> <p>The Professional edition fully supports code import/export and synchronization of model elements with source code; it enables reverse engineering of SQL Server; MS Access 97, 2000 and 2003; and Oracle 9i, 10g or 11g databases</p> <p>Support for MDG Technologies and MDG Link (sold separately) is included with the Professional version</p> <p>The shared repository available in the Professional edition is restricted to the .EAP file format (JET database)</p>	
<p>Desktop Edition</p> 	<p>The Desktop edition is targeted at single analysts and developers producing UML analysis and design models</p> <p>It provides facilities for:</p> <ul style="list-style-type: none"> • UML modeling • XML import/export • Document generation • Version control integration and 	

Topic	Detail	See also
	<ul style="list-style-type: none"> Profile/metamodel extensibility. 	

1.2.3 The Read-only 'Lite' Edition

Enterprise Architect Lite is a **free**, *read-only* edition of Enterprise Architect that enables people such as project sponsors to review the project without making any changes.

Topic	Detail	See also
Further Access	<p>Users of Enterprise Architect Lite also have wider access to</p> <ul style="list-style-type: none"> The Team Review, where readers can create and respond to posts, and link their comments to elements The Source Code Viewer, where readers can open and edit external source code files, debug code, and configure and run Analyzer scripts The File menu, where readers can copy the project or create a shortcut to access it The Relationship Matrix, where readers can export the matrix contents to a CSV file to be opened in a spreadsheet application The Default Hours tab to review project metrics. <p>You can download the Enterprise Architect Lite edition (as the Enterprise Architect Viewer) from the Sparx Systems website</p>	<p>Team Review Tools ^[216]</p> <p>The Source Code Viewer ^[1408]</p> <p>File menu ^[74]</p> <p>Relationship Matrix Options ^[501]</p> <p>Default Hours ^[404]</p> <p>http://www.sparxsystems.com/products/ea/downloads.html.</p>
Other Read-Only Options	<p>You can also make your model available to others in a read-only format by:</p> <ul style="list-style-type: none"> Generating an HTML report on the model, which can be published on the web with read-only access 	<p>HTML Reports ^[1802]</p>

1.3 Formal Statements



Please take the time to read the following legal statements concerning Sparx Systems Enterprise Architect:

Topic	Link
Software Copyright Notice	Software Copyright Notice ^[19]
Enterprise Architect End User Licensing Agreement	Enterprise Architect End User Licensing Agreement ^[20]
Acknowledgement of Trademarks	Acknowledgement of Trademarks ^[23]

Spark Systems would also like to gratefully [acknowledge contributions](#) ^[25] to the development of Enterprise Architect.

1.3.1 Copyright Notice

Copyright © 1998 - 2011 Sparx Systems Pty. Ltd. All rights reserved.

The software contains proprietary information of Sparx Systems Pty Ltd. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. Please read the [license agreement](#) ^[20] for full details.

Due to continued product development, this information can change without notice. The information and intellectual property contained herein is confidential between Sparx Systems and the client and remains the exclusive property of Sparx Systems. If you find any problems in the documentation, please report them to us in writing. Sparx Systems does not warrant that this document is error-free. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Sparx Systems. Licensed users are granted the right to print a single hardcopy of the user manual per licensed copy of the software, but may not sell, distribute or otherwise dispose of the hardcopy without written consent of Sparx Systems.

Sparx Systems Pty. Ltd.

7 Curtis St,
Creswick, Victoria 3363,
AUSTRALIA

Phone: +61 (3) 5345 1140
Fax: +61 (3) 5345 1104

Support Email: support@sparxsystems.com
Sales Email: sales@sparxsystems.com

Website: www.sparxsystems.com

Scintilla and SciTE

Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org> All Rights Reserved.

Permission to use and distribute this (Scintilla) software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

1.3.2 End User License Agreement

Enterprise Architect Modeling Tool Version 9.0

Desktop, Professional, Corporate, Business and Software Engineering, Systems Engineering & Ultimate Editions

Copyright (C) 1998-2011 Sparx Systems Pty Ltd. All Rights Reserved

IMPORTANT- READ CAREFULLY: This End User License Agreement ("EULA") is a legal agreement between YOU as Licensee and SPARX SYSTEMS ("SPARX") for the SOFTWARE PRODUCT identified above. By installing, copying, or otherwise using the SOFTWARE PRODUCT, YOU agree to be bound by the terms of this EULA. If YOU do not agree to the terms of this EULA, promptly return the unused SOFTWARE PRODUCT to the place of purchase for a full refund.

The copyright in the SOFTWARE PRODUCT and its documentation is owned by Sparx Systems Pty Ltd A.C.N 085 034 546. Subject to the terms of this EULA, YOU are granted a non-exclusive right for the duration of the EULA to use the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

Your use of this software indicates your acceptance of this EULA and warranty.

DEFINITIONS

In this End User License Agreement, unless the contrary intention appears:

- "ACADEMIC EDITION" means an edition of the SOFTWARE PRODUCT purchased for educational purposes at an academic discount price.
- "EULA" means this End User License Agreement.
- "SPARX" means Sparx Systems Pty Ltd A.C.N 085 034 546.
- "Licensee" means YOU, or the organization (if any) on whose behalf YOU are taking the EULA.
- "Registered Edition of Enterprise Architect" means the edition of the SOFTWARE PRODUCT which is available for purchase from the web site: http://www.sparxsystems.com/ea_purchase.htm.
- "SOFTWARE PRODUCT" or "SOFTWARE" means Enterprise Architect, UML Case Tool, Desk Top, Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, which includes computer software and associated media and printed materials, and may include online or electronic documentation.
- "Support Services" means email based support provided by SPARX, including advice on usage of Enterprise Architect, investigation of bugs, fixes, repairs of models if and when appropriate, and general product support.
- "SPARX support engineers" means employees of SPARX who provide on-line support services.
- "Trial edition of Enterprise Architect" means the edition of the SOFTWARE PRODUCT which is available free of charge for evaluation purposes for a period of thirty (30) days.
- "EALITE" means the LITE version of Enterprise Architect that is distributed free of charge as a read-only viewer of .EAP files.

GRANT OF LICENSE

In accordance with the terms of this EULA YOU are granted the following rights:

- a) To install and use one copy of the SOFTWARE PRODUCT or, in its place, any prior version for the same operating system, on a single computer. As the primary user of the computer on which the SOFTWARE PRODUCT is installed, YOU may make a second copy for your exclusive use on either a home or portable computer.
- b) To store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT over an internal network. If YOU want to increase the number of users entitled to concurrently access the SOFTWARE PRODUCT, YOU must notify SPARX and agree to pay an additional fee.
- c) To make copies of the SOFTWARE PRODUCT for backup and archival purposes only.

EVALUATION LICENSE

The Trial version of Enterprise Architect is not free software. Subject to the terms of this agreement, YOU are hereby licensed to use this software for evaluation purposes without charge for a period of thirty (30) days.

Upon expiration of the thirty (30) days, the SOFTWARE PRODUCT must be removed from the computer. Unregistered use of Enterprise Architect after the 30-day evaluation period is in violation of Australian, U.S. and international copyright laws.

SPARX may extend the evaluation period on request and at their discretion.

If YOU choose to use this software after the 30 day evaluation period a license must be purchased (as described at http://www.sparxsystems.com/ea_purchase.htm). Upon payment of the license fee, YOU will be sent details on where to download the registered edition of Enterprise Architect and will be provided with a suitable software 'key' by email.

EALITE

Subject to the terms of this Agreement [EALITE](#) may be installed on any machine indefinitely and free of charge. There are no fees or Sparx support services in relation to EALITE.

ADDITIONAL RIGHTS AND LIMITATIONS

YOU hereby undertake not to sell, rent, lease, translate, adapt, vary, modify, decompile, disassemble, reverse engineer, create derivative works of, modify, sub-license, loan or distribute the SOFTWARE PRODUCT other than as expressly authorized by this EULA.

YOU further undertake not to reproduce or distribute license key-codes except under the express and written permission of SPARX.

If the Software Product purchased is an Academic Edition, YOU ACKNOWLEDGE THAT the license is limited to use in an educational context, either for self-education or use in a registered teaching institution. The Academic Edition may not be used to produce commercial software products or be used in a commercial environment, without the express written permission of SPARX.

ASSIGNMENT

YOU may only assign all your rights and obligations under this EULA to another party if YOU supply to the transferee a copy of this EULA and all other documentation including proof of ownership. Your license is then terminated.

TERMINATION

Without prejudice to any other rights, SPARX may terminate this EULA if YOU fail to comply with the terms and conditions. Upon termination YOU or YOUR representative shall destroy all copies of the SOFTWARE PRODUCT and all of its component parts or otherwise return or dispose of such material in the manner directed by SPARX.

WARRANTIES AND LIABILITY

WARRANTIES

SPARX warrants that the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and any Support Services provided by SPARX shall be substantially as described in applicable written materials provided to YOU by SPARX, and SPARX support engineers will make commercially reasonable efforts to solve any problems associated with the SOFTWARE PRODUCT.

EXCLUSIONS

To the maximum extent permitted by law, SPARX excludes, for itself and for any supplier of software incorporated in the SOFTWARE PRODUCT, all liability for all claims, expenses, losses, damages and costs made against or incurred or suffered by YOU directly or indirectly (including without limitation lost costs, profits and data) arising out of:

- YOUR use or misuse of the SOFTWARE PRODUCT
- YOUR inability to use or obtain access to the SOFTWARE PRODUCT
- Negligence of SPARX or its employees, contractors or agents, or of any supplier of software incorporated in the SOFTWARE PRODUCT, in connection with the performance of SPARX' obligations under this EULA, or
- Termination of this EULA by either party for any reason.

LIMITATION

The SOFTWARE PRODUCT and any documentation are provided "AS IS" and all warranties whether express, implied, statutory or otherwise, relating in any way to the subject matter of this EULA or to this EULA generally, including without limitation, warranties as to: quality, fitness; merchantability; correctness; accuracy; reliability; correspondence with any description or sample, meeting your or any other requirements; uninterrupted use; compliance with any relevant legislation and being error or virus free are excluded. Where any legislation implies in this EULA any term, and that legislation avoids or prohibits provisions in a contract excluding or modifying such a term, such term shall be deemed to be included in this EULA. However, the liability of SPARX for any breach of such term shall if permitted by legislation be limited, at SPARX's option to any one or more of the following upon return of the SOFTWARE PRODUCT and a copy of the receipt:

- If the breach relates to the SOFTWARE PRODUCT:
 - the replacement of the SOFTWARE PRODUCT or the supply of an equivalent SOFTWARE PRODUCT
 - the repair of such SOFTWARE PRODUCT
 - the payment of the cost of replacing the SOFTWARE PRODUCT or of acquiring an equivalent SOFTWARE PRODUCT, or
 - the payment of the cost of having the SOFTWARE PRODUCT repaired.
- If the breach relates to services in relation to the SOFTWARE PRODUCT:
 - the supplying of the services again, or
 - the payment of the cost of having the services supplied again.

TRADEMARKS

All names of products and companies used in this EULA, the SOFTWARE PRODUCT, or the enclosed documentation may be trademarks of their corresponding owners. Their use in this EULA is intended to be in compliance with the respective guidelines and licenses.

Windows®, Windows 98, Windows NT, Windows ME, Windows XP, Windows Vista, Windows 2000 and Windows 2003 Server are trademarks of Microsoft®.

GOVERNING LAW

This agreement shall be construed in accordance with the laws of the Commonwealth of AUSTRALIA, in the state of Victoria.

1.3.3 Trademarks

Sparx Systems acknowledge the following trademarks and registered trademarks, which are used throughout the Enterprise Architect documentation.

Topic	Trademarks	Registered Trademarks
Microsoft Corporation	ActiveX™ Microsoft Office™ Microsoft Word™ VB.NET™	Access® Active Directory® Excel® IntelliSense® JScript® Microsoft® MS Access® MS Access 2007® MSDE® SQL Server® Visio® Visual Basic® Visual SourceSafe® Visual Studio® Win32® Windows® Windows Vista®
The Object Management Group (OMG)	OMG™ Object Management Group™ The CORBA logo ORB™ Object Request Broker™ The CORBA Academy design	The OMG Object Management Group logo The Information Brokerage® CORBA® CORBA Academy® IIOP® XMI®

Topic	Trademarks	Registered Trademarks
	OMG Interface Definition Language™ IDL™ CORBAservices™ CORBAfacilities™ CORBAmed™ CORBAnet™ Unified Modeling Language™ MOF™ CWM™ BPMN™ SoaML™ Model Driven Architecture™ MDA™ OMG Model Driven Architecture™ OMG MDA™	UML® The UML Cube logo
The Open Group		Archimate®
Methodologies Corporation	SOMF™	
Embarcadero Technologies Inc	InterBase™	Delphi®
Oracle Corporation	MySQL™	Oracle® Java®
Python Software Foundation	Python™	
Linux Mark Institute	Linux™	
Innbase Oy	InnoDB™	
Firebird Foundation Incorporated		FireBird®
Ingres Corporation	Ingres™	
IBM	DB2 Informix	
Sybase Inc	Sybase Central™	Adaptive Server® Adaptive Server® Anywhere Adaptive Server® Everywhere SQL Anywhere® Sybase®

Topic	Trademarks	Registered Trademarks
Progress Software Corporation		OpenEdge® Progress OpenEdge®

1.3.4 Acknowledgements

Some parts of this application include code originally written by various authors and modified for use in Enterprise Architect.

Name	Detail	Link
Marquet Mike	Print listview contents	mike.marquet@altavista.net
Davide Pizzolato	CXImage Library © 7-Aug-2001	ing.davide.pizzolato@libero.it
Neil Hodgson	Scintilla editor © 1998-2003	neilh@scintilla.org
Others	Many thanks to all those who have made suggestions, reported bugs, offered feedback and helped with the beta-testing of Enterprise Architect Your help has been invaluable	

1.4 Order Enterprise Architect

Enterprise Architect is designed, built and published by Sparx Systems, and available through the Sparx Systems website as either a:

- Free, 30-day trial version (although you can extend the evaluation period)
- Registered purchase version that provides the facilities of the edition you require, as unlocked by the license keys that you purchase from Sparx Systems

The trial version of Enterprise Architect is identical to the registered version with the exception that all diagrams are output to files with an embedded watermark. The trial software stops working after the trial period has elapsed.

On purchase of a suitable license or licenses, the registered version is made available for download.

The latest information on pricing and purchasing is available on the Purchase and Pricing page of the Sparx Systems website, and from Sparx Sales.

Purchase Options

- On-line using a secure credit-card transaction
- Fax
- Check or equivalent
- Bank transfer

Learn More:

- [The Trial Version](#)¹³
- [Editions Available](#)¹⁴
- sales@sparxsystems.com
- [Pricing and Purchase Options](#)

1.5 Installation

Enterprise Architect is distributed as a single executable setup file (.exe). After you execute this file, Enterprise Architect is immediately available to create projects as .EAP files.

The latest evaluation and registered versions of Enterprise Architect are always available from the Sparx Systems website.

The registered version is available through the registered user area of the web site, which requires a username and password to access. These are provided upon purchase of a license.

Topic	Detail	See also
System Requirements	The system requirements for installing Enterprise Architect are defined on the Enterprise Architect System Requirements page of the Sparx Systems website.	Enterprise Architect System Requirements
Windows Vista	<p>Under Windows Vista (with User Account Control turned on) an application starts with only Standard permissions, regardless of what level of authority the current user has. As a result, an installer run normally with an Admin account under Vista only has Standard privileges and either is not able to write to certain critical areas of the registry/file system, or redirects the write requests to a per-user virtualized registry/file system.</p> <p>Sparx Systems recommend that if you are installing on Windows Vista, always run the Enterprise Architect installer with Administrator privileges (right-click on the downloaded installer icon and select the Run as administrator menu option).</p>	
Install Enterprise Architect	<p>Run the Enterprise Architect setup program. Generally you can accept all the default options without change.</p> <p>To place Enterprise Architect in a directory other than the default, enter the name of the destination when prompted.</p> <p>You might be prompted to restart your computer when the installation completes. Although this is not always necessary (if you already have the components Enterprise Architect requires installed on your computer), you should restart just to be certain.</p>	
Installation on Linux	<p>If you intend to run Enterprise Architect on Linux, refer to the page "<i>Installing Enterprise Architect inside Wine</i>", on the Sparx Systems website.</p> <p>To support the Scripting facility in the Corporate and extended editions of Enterprise Architect, you must also install Internet Explorer 6.0 or later revisions.</p>	Installation inside Wine
Using a Third Party DBMS as Model Repository	If you plan to use SQL Server, MySQL, PostgreSQL, Access 2007, Sybase Adaptive Server Anywhere or Oracle 9i, 10g or 11g as a model repository, then you must use the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions of Enterprise Architect.	Corporate Edition Resources Page Trial Corporate Edition Resources Page

Topic	Detail	See also
	<p>You also require additional files and supplementary installation processes. Please note that installation and maintenance of these database management systems is not covered under the support agreement.</p> <p>Users planning to use SQL Server, MySQL, PostgreSQL, Sybase Adaptive Server Anywhere, Access 2007 or Oracle 9i, 10g or 11g as their model repository can access scripts that create the required data structures for the choice of DBMS, from the Sparx Systems website. To access these scripts, please follow the links at right.</p>	

Notes:

- Enterprise Architect requires **Read/Write** access to the program files directory where Enterprise Architect has been installed

1.6 Register a Full License

The **trial** version of Enterprise Architect available for download is an evaluation version only. For the **full** version you must first purchase one or more *licenses*.

The license supplied determines which edition (Enterprise Architect Desktop, Professional, Corporate, Business and Software Engineering, System Engineering or Ultimate) is activated after installation.

The Corporate, Business and Software Engineering, System Engineering and Ultimate editions can be activated by either a private license key or a shared (floating) license key. Shared license keys, for a user population that includes temporary users, are checked out of and in to a Sparx Systems key store.

Use To:

- Obtain and activate the registered version of Enterprise Architect

How To:

Step	Action	See also
1	Purchase one or more licenses Once you have paid for a licensed version of Enterprise Architect, you receive (via email or other suitable means): <ul style="list-style-type: none"> • a license key or keys • a web site address from which to download the full version • if you have purchased a Sparx Systems key store, a web site address from which to download the key store 	Editions Available <small>[14]</small>
2	Save the license key(s) and download the latest full install package from the address supplied	
3	Run the setup program to install the full version	
4	Open Enterprise Architect from the Start Menu or desktop icon The License Management dialog displays	
5	Click on the Add Key button The Add Registration Key dialog displays; register your key	Add License Key <small>[2096]</small>
6	Click on the OK button The full version is now activated on your PC, and Enterprise Architect displays the message: Registration succeeded! Thank you for purchasing Enterprise Architect <type> Edition	

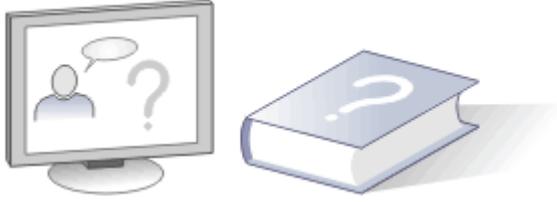
Notes:

- Private license keys and shared license keys have different formats, so you cannot use one in place of the other.

Learn More:

- [Upgrade an Existing License](#) [2099]
- [Register Add-In](#) [2107]

1.7 Help and Support



Enterprise Architect has three main help and information systems to assist you in using the product:

- Learning Center
- Enterprise Architect Help
- The Sparx Systems website.

In addition Sparx Systems recommend that you fully explore the sample project supplied with Enterprise Architect. It assists you in learning to use Enterprise Architect and offers tips on getting the most out of Enterprise Architect's features. Click on the **EAExample** option on the Enterprise Architect Start Page.

Topic	Detail	See also
Learning Center	<p>The Enterprise Architect Learning Center provides context-sensitive guidance, tools, demonstrations and other online resources to help you understand any area of Enterprise Architect that you are interested in.</p> <p>The Learning Center automatically displays on the right of the screen when you first open Enterprise Architect, showing the Getting Started topics. You can select other task areas by clicking on the drop-down arrow in the toolbar.</p>	Learning Center ^[72]
Enterprise Architect Help	<p>Enterprise Architect Help provides comprehensive documentation of Enterprise Architect and covers every aspect and facility of the product. To access Help within Enterprise Architect:</p> <ul style="list-style-type: none"> • Click on the Help icon () in the various toolbars • Select the Help Help Contents menu option • Click on the Help button on a dialog (for Help specific to that dialog). <p>Enterprise Architect Help is extensive; if you cannot quickly locate the topic you require in the online contents list, you can use one of two search facilities:</p> <ul style="list-style-type: none"> • Click on the Index tab, type in a keyword or key phrase appropriate to the subject you require help for, and press (Enter); double-click on the appropriate index item • Click on the Search tab, type in a word or phrase to search for, and click on the List Topics button; double-click on the required topic. <p>The Enterprise Architect Help is also available separately from the product, in different formats.</p>	Available Help File Formats ^[31]
Sparx Systems Web Site	<p>The Sparx Systems web site is also extensive, and provides information and announcements concerning the company and its full range of products, as well as tutorials, white papers, templates and solutions.</p>	Sparx Systems Web Site Enterprise Architect User Forum

Topic	Detail	See also
	<p>It also provides a user forum, community site (see below) and support network; Sparx Systems are highly responsive to user feedback and requirements, and the web site enables rapid communication concerning problems, solutions and enhancements.</p> <p>You can access the web page and user forum within Enterprise Architect from the View More Project Tools Internal Web Browser menu option, and through the Online Resources topics in the Learning Center.</p>	Learning Center ^[72]
Community Site	<p>The Sparx Systems website also hosts the Sparx Systems Enterprise Architect Community Site. This is a central location for the Enterprise Architect community to publish resources and share experiences.</p> <p>From this site you can download the latest news, tutorials, resources, best practices, tips, techniques and user-generated content for Enterprise Architect. You can also, as a registered author, contribute content and share your expertise with the wider community.</p>	Sparx Systems Enterprise Architect Community Site

Learn More:

- [Contacting Sparx Support](#)^[31]

1.7.1 Available Helpfile Formats

You can access the latest Enterprise Architect help files from the following locations:

Format	Link
.CHM	www.sparxsystems.com/bin/EA.chm
.CHM inside a .ZIP file:	www.sparxsystems.com/bin/EAHelp.zip
.PDF	www.sparxsystems.com/bin/EASUserGuide.pdf
.HTML	www.sparxsystems.com/EASUserGuide/index.html

Version and release date information for the help files can be found at:

Topic	Link
Help file	www.sparxsystems.com/ea_downloads.htm#Helpfiles ,
Help file(registered users).	www.sparxsystems.com/registered/reg_ea_down.htm#Helpfiles

1.7.2 Support

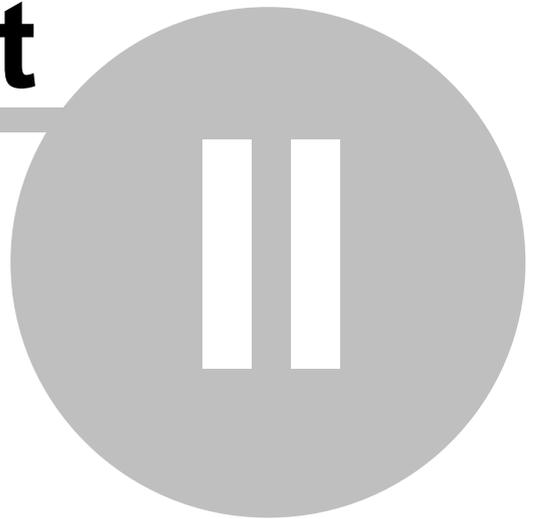
Technical support for Enterprise Architect is available to registered users. Responses to support queries are sent by email.

Sparx Systems endeavors to provide a rapid response to all product-related questions or concerns.

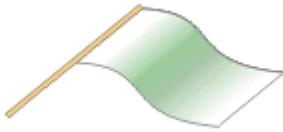
- Registered users can lodge a support request, by visiting: http://www.sparxsystems.com/registered/reg_support.html

- Trial users can contact Sparx Systems with questions regarding their evaluation at: support@sparxsystems.com
- An online user forum is also available for your questions and perusal, at <http://www.sparxsystems.com/cgi-bin/yabb/YaBB.cgi>

Part



2 Getting Started



This guide helps you to understand the options available when you start Enterprise Architect, and to quickly gain an understanding of how to use these options to create models.

Topic	Link
After starting Enterprise Architect, use the Quick Start tutorial to immediately create a project	A Quickstart Tutorial ^[37]
Read about work areas applicable to certain Project Roles	Project Roles ^[57]
Review the Enterprise Architect User Interface, or workspace	Enterprise Architect User Interface ^[69]

At various points throughout the Enterprise Architect Help, there are further Quick Start topics and sections to help you use the system immediately to experiment with a feature of Enterprise Architect. Use the Help Index tab and search for *Quick Start* to locate these topics.

Learn More:

- [Start Enterprise Architect](#) ^[35]

2.1 Basics



When you install Enterprise Architect on your computer, a new program folder called *Enterprise Architect* is created in your **Start** menu (unless you changed the default name during installation).

You can start Enterprise Architect from the icon created on your Windows desktop during installation, or alternatively:

1. Open the Windows **Start** menu
2. Locate the Enterprise Architect program folder
3. Select **Enterprise Architect**

After a short pause, the Start Page displays. From this page you can:

Topic	Link
Open a project file (.EAP file)	Open a project file ^[138]
Create a new project (.EAP file)	Create a new project ^[145]
Connect to a DBMS repository(Corporate and extended editions)	Connect to a DBMS repository ^[161]

Notes:

- By default, when you install Enterprise Architect, an empty 'starter' project called *EABase.EAP* is installed, as well as an example project named *EAExample.EAP*. We recommend that new users select the EAExample file and explore it in some detail while you become familiar with UML and software engineering using Enterprise Architect
- Enterprise Architect.EAP files default to use Jet 3.5 as the database engine. This does not support unicode character sets. If you want to use unicode character sets (for example, to provide user interface texts in languages other than your Windows-defined native language), you must either upsize to a DBMS repository or set JET 4.0 as the database engine. Resetting the database engine ensures compatibility with .EAP files that support unicode character sets and that are in turn compatible with versions of MS Access later than Access 97

If your .EAP project is not in a Jet 4.0 database, you should also download a copy of the Jet 4.0 EABase model from the Sparx Systems website, and do an EAP to EAP transfer of your model into the Jet 4.0 file

Learn More:

- [Start Page](#)^[70]
- [A Quick Start Tutorial](#)^[37], to begin a guided exploration of Enterprise Architect immediately
- [Common Tasks](#)^[36], to get an overview of the kind of work you might perform with Enterprise Architect
- [Upsize to a DBMS Repository](#)^[148]
- [General Options](#)^[422] to set JET 4.0 as database engine
- [Sparx Systems website](#) for the Jet 4.0 EABase model
- [Perform a Data Transfer](#)^[343]

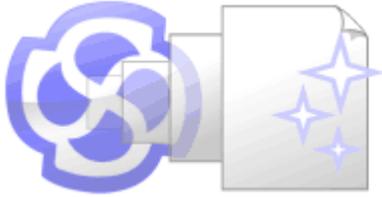
2.2 Common Tasks

Enterprise Architect is a powerful modeling tool that can be used by project managers, business analysts, developers and testers alike to build business, real-time and software systems.

The following list outlines some of the common tasks performed by Enterprise Architect and describes some of the tools that you might be required to use.

Topic	Detail
Manage Projects Effectively	Enterprise Architect provides all the tools necessary for the planning, execution and successful completion of a project <ul style="list-style-type: none"> • Create a shared vision of your project, improve communication and help team members agree upon the design of a proposed system • Evaluate project risk factors, report any changes or defects and develop strategies to tackle potential problems • Help estimate the amount of time and effort required to complete a project and maximize staff utilization • Use Auditing, Project Baselines and User Access Controls to better manage change in a collaborative environment
Analyze Strategic Business Needs	A Business Analyst is responsible for gathering requirements, modeling potential solutions and evaluating the business needs of an organization <ul style="list-style-type: none"> • Build a coherent picture of how a business operates through requirements gathering, Use Case analysis and modeling business rules • Help achieve business process improvement and implement change using tools such as Strategic Models, Use Case Scenarios, Business Rule Models, Flow Charts, Auditing and the Team Review window
Build and Develop Systems	The developer is responsible for implementing a solution using a range of tools <ul style="list-style-type: none"> • Bring the power and benefits of Enterprise Architect into your favorite Integrated Development Environment • Use Enterprise Architect's in-built support for writing code, including syntax highlighting, line numbering, auto completion, bookmarking and automatic indentation • Create breakpoints to step through code execution, examine variables and view error messages via the debug output • Apply Visual Execution Analysis to identify costly function calls, explain system behavior and establish the sequence of events that occur immediately prior to system failure • Use Enterprise Architect's support for round trip engineering and synchronizing source code with a corresponding model • In addition to many popular programming languages, create powerful database solutions that automatically generate DDL scripts for more than 10 different Database Management Systems
Test and Investigate	Quality Assurance teams test software to identify defects, verify that it satisfies all requirements and ensure that it produces expected results <ul style="list-style-type: none"> • In addition to the integrated JUnit and NUnit testing capabilities, Enterprise Architect enables you to create and manage test scripts for model elements, covering unit, integration, scenario, system and acceptance tests • Visualize the execution of code to better understand how applications work and the sequence of events that leads to program failure

2.3 A Quickstart Tutorial



Welcome to Enterprise Architect!

This quick-start tutorial is designed to help you get up to speed with Enterprise Architect.

As you read through this tutorial, it is recommended that you have Enterprise Architect open so that you can try out the tasks described. By the end you should be able to begin modeling your own business/software projects with Enterprise Architect.

The tutorial guides you through a number of important basic tasks:

Topic	Link
Creating a new Project	Creating a New Project ^[37]
Adding Views to your Model	Views ^[38]
Adding Packages to your Model	Packages ^[39]
Adding Diagrams	Adding Diagrams ^[40]
Adding elements to your model packages	Elements ^[41]
Creating links between model elements	Links ^[42]
Refining the model	Refining the Model ^[44]

Throughout the descriptions there are hyperlinks to more detailed information on a range of topics.

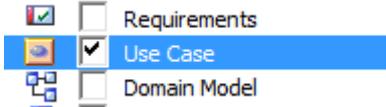
2.3.1 Create Project Scenario

A project is a single file or repository-based store for one or more Models.

The first step in getting started with Enterprise Architect is to either open an existing project, or create a new one.

In this example we create a new file-based project and add some template-based models to kick-start our modeling.

Step	Action	Result
1	Start Enterprise Architect	The Open Project dialog displays (If the dialog does not appear, press (Ctrl+O) to invoke it)
2	Click the New Project button to select a suitable location and name for your new project	The standard Windows file browser dialog displays. File-based Enterprise Architect projects are

Step	Action	Result
		named with a .EAP extension
3	Locate a suitable folder for your project and enter a distinctive name Press the Save button to continue	Enterprise Architect creates a new project file and places it in the specified location The project is then opened automatically and the New Model Wizard displays
4	In the left hand column ensure that Default is selected - and in the right hand side tick the Use Case View check box  Click on the OK button	The Model Wizard automatically creates a new Use Case model for you, with an initial diagram, some notes and default elements to help get you started At this stage all your changes have been saved to file and do not require further action

Your new project has now been created, with a simple Use Case View added for you to explore and tailor to your requirements. You can re-open your project at any time by double-clicking on it in a file browser. It should also appear in your **Recent Projects** list on the Start page.

2.3.2 Add a View To Your Model

A View is a top level package, within a model. It can be categorized by purpose using different icon types, such as Use Case View, Component View or Deployment View. Views are used to contain packages, diagrams and elements - the building blocks of your model.

Use To:

- Create a new View in your project

How To:

Step	Action	See also
1	Select a model root node in the Project Browser, then either: <ul style="list-style-type: none"> • Click the Add a Package button  on the Project Browser toolbar. • Select Project New Package • Right-click the model root node, from the context menu select New View • Press (Ctrl + W) The Create New View dialog displays	Views ^[530]
2	Enter an appropriate name for the new View	
3	Click the appropriate radio button to select an icon for the View	
4	Click on the OK button The Create New View dialog closes The new View is created as a child of the selected model root node	

Learn More:

- [Add a package to your model](#) ^[39]
- [Create a new diagram](#) ^[40]
- [Add an element](#) ^[41]
- [Read more about views](#) ^[530]

Notes:

- There are six types of View, which represent conventional ways of categorizing the purpose of a Model View; these Views represent different structural or behavioral aspects of the same model

2.3.3 Add Packages

A Package is a container of model elements, and is displayed in the Project Browser as a 'folder' icon. A Package is required to hold the model building blocks, such as diagrams, elements and other packages.

So, to begin developing your Model, you must create an initial package for your first diagrams.

Use To:

- Add new packages to your model

How To:

Step	Action	See also
1	<p>Select a Package or View in the Project Browser, then either:</p> <ul style="list-style-type: none"> • Click the Add a Package button  on the Project Browser toolbar. • Press (Ctrl + W) • Select Project New Package • Right-click the Package or View and, from the context menu, select Add Add Package <p>The New Model Package dialog displays</p>	
2	Enter a suitable name for the package	
3	Optionally, clear the check mark from Automatically add new diagram	
4	<p>Click on the OK button</p> <p>The New Model Package dialog closes</p> <p>A new package is created as a child of the selected package or View</p>	

Notes:

- When the **Automatically add new Diagram** option is selected, Enterprise Architect also creates a default diagram within the new package, to speed your development

Learn More:

- [Creating Diagrams](#) ^[40]
- [Add Elements](#) ^[41]
- [Move Components](#) ^[44]

- [Learn more about the Project Browser](#)^[44]

2.3.4 Creating Diagrams

A diagram is a visual representation of the elements of your model, their attributes and characteristics, and how they are connected and/or interact with each other.

Different diagram types show different aspects of the model and the relationships between elements.

Use To:

- Add new diagrams to your model

How To:

Step	Action	See also
1	<p>Select a package or View in the Project Browser, then either:</p> <ul style="list-style-type: none"> • Click on the Add a Diagram button  on the Project Browser toolbar • Press (Ctrl + Insert) • Select Project New Diagram • Right-click the package or View and, from the context menu, select Add Add Diagram <p>The New Diagram dialog displays</p>	
2	<p>Enterprise Architect provides a default diagram name, that is the same name as the parent package</p> <p>Update the diagram name as required</p>	
3	<p>Select a category of diagrams in the left hand pane</p> <p>The list of available diagram types in the right hand pane is updates according to your selection</p>	
4	<p>Select the type of diagram to add from the right hand pane</p>	
5	<p>Click on the OK button</p> <p>The New Diagram dialog closes</p> <p>A new diagram is created as a child of the currently selected package</p> <p>The new diagram is opened in the Diagram View, ready for editing</p>	

Notes:

- When you create a package, if you leave the **Automatically add new Diagram** option selected, the New Diagram dialog displays automatically

Learn More:

- [Add Elements](#)^[41]
- [Move Elements](#)^[44]
- [Learn more about UML diagrams](#)^[795]

2.3.5 Creating New Elements on Diagrams

Models are constructed from elements, each of which has its own meaning, rules and notation. Generally, when you create a new element in your model, you want to use that element on a diagram. The simplest way to do this is to create the element directly on the diagram.

Use To:

- Create new elements on diagrams in your model

How To:

Step	Action	See also
1	Ensure that the Diagram Toolbox is visible To display the Diagram Toolbox, either: <ul style="list-style-type: none"> • Select View Diagram Toolbox • Press (Alt + 5) 	
2	Ensure that the diagram on which the elements are to be created is open in the Diagram View To open the diagram, double-click on the diagram name in the Project Browser The diagram opens and the Toolbox is updated to display the categories of elements and relationships that are applicable to that type of diagram	Diagram View ^[540]
3	Click on the appropriate icon in the Toolbox to select the type of element to create The element type is highlighted in the Toolbox	
4	Click on the diagram at the location where you want to place the element The new element is created as a child of the package that contains the diagram, and is placed onto the diagram at the cursor position The element Properties dialog displays	
5	Use the Properties dialog to define characteristics of the element as required	Properties Dialog ^[660]
6	Click on the OK button The Properties dialog closes	

Notes:

- You can also drag or paste existing elements onto a diagram from the Project Browser
- If you are creating several elements of one type, after creating the first just press (**Shift+F3**) or (**Ctrl +Click**) to create the next element of that type

Learn More:

- [Paste From the Project Browser](#) ^[576]
- [Add connectors](#) ^[42]
- [The Quick Linker](#) ^[622]
- [Move elements](#) ^[44]
- [UML elements](#) ^[864]

2.3.6 Adding Connectors

Connectors define specific relationships between specific elements, so you usually create them directly on the diagram by dragging the required relationship type from the Diagram Toolbox. As for elements, the Toolbox automatically presents the connector or relationship types appropriate to the type of diagram.

Use To:

- Define a relationship between two model elements on a diagram

How To:

Step	Action	See also
1	Ensure that the Diagram Toolbox is visible To display the Diagram Toolbox, either: <ul style="list-style-type: none"> • Select View Diagram Toolbox or • Press (Alt + 5) 	
2	Ensure that the diagram containing the elements to be connected, is open in the Diagram View To open the diagram, double-click on the diagram name in the Project Browser The selected diagram opens and the Toolbox is updated to display the categories of elements and relationships that are applicable to that diagram type	Diagram View ^[540] Creating Elements on Diagrams ^[41]
3	Click on the required connector in the Toolbox to select the type of connector to draw The connector type is highlighted in the Toolbox	
4	Click on the source element in the relationship, then drag across to the target element The selected connector is drawn between the two elements The connector Properties dialog displays	
5	Use the Properties dialog to define the characteristics of the connector as required	Properties Dialog ^[660]
6	Click on the OK button The Properties dialog closes	

Notes:

- If you are creating several connectors of one type, after creating the first just click on the appropriate source element and press (**F3**) to create the next connector of that type
- As you drag a connector, you can press (**Shift**) to create a bend in the connector; if necessary, you can put several bends in the connector line, pressing (**Shift**) every time you want to change direction
- To roll back the bends, keep holding the left mouse button down and press (**Backspace**) as many times as is necessary
- To find out more about the type of connector you have dragged on to a diagram, right-click on the connector and select the **UML Help** menu option, which displays a Help page on the connector type

Learn More:

- [Quick Linker - Create Connectors](#) ^[624]
- [Create Connector in Project Browser](#) ^[748]
- [UML Connections](#) ^[966]

2.3.7 Modify Properties

When you create an element and connect it to another element, you usually have to define various characteristics of both the element and the connector to identify the purpose and function they represent. You do this using a Properties dialog.

When you create elements, Enterprise Architect automatically names and numbers them by type - for example, Class1, Class2 - so you should at least change the **Name** field to more easily identify each element. Enterprise Architect does not automatically name connectors, but for many connector types you should provide a name that describes the purpose of the connection.

Use To:

- Modify the characteristics of a model element or connector

How To:

Step	Action	See also
1	Either: <ul style="list-style-type: none"> • Double-click on an element or connector in the diagram • Right-click an element in the Project Browser and, from the context menu, select Properties • Right-click a connector in a diagram and, from the context menu, select Properties The appropriate Properties dialog displays	
2	Modify the element or connector characteristics as required	Element Properties Dialog ^[660] Connector Properties Dialog ^[756]
3	Click on the OK button The Properties dialog closes The modifications are saved; the Project Browser and diagrams that contain the element are updated as necessary	

Notes:

- Enterprise Architect is initially configured to display the Properties dialog automatically when you create an element or connector, but it is easy (and often convenient) to turn the dialog display off
- If the default display has been turned off, you can display the Properties dialog by either double-clicking on the element or connector in the diagram or by right-clicking on it in the Project Browser and selecting the **Properties** menu option

Learn More:

- [Element Properties Dialog](#) ^[660]
- [Connector Properties Dialog](#) ^[756]

2.3.8 Moving Elements, Diagrams and Connectors

You have created a project containing packages, diagrams and elements, and you have connected the elements. You might have arranged your components in the wrong project structure. How do you change where things are?

Topic	Link
Changing the Project Browser list order - re-arrange the order of items in the Project Browser	Move Elements Within a Package ^[44]
Moving elements between packages - move diagram, elements and child packages, from one parent package to another	Move Elements Between Packages ^[45]
Moving elements on a diagram - re-arrange the position of diagram elements	Move Elements in a Diagram ^[45]
Moving elements between diagrams - cut an element from one diagram and paste it onto another	Move Elements Between Diagrams ^[46]
Moving connectors on a diagram - re-link a source element to a different target element	Move Connectors on a Diagram ^[48]

Learn More:

- [Delete Components](#) ^[48]
- [Arrange Connectors](#) ^[744]
- [Order Package Contents](#) ^[455]

2.3.8.1 Move Elements Within a Package

In the Project Browser, the contents of a package are listed in the order: diagrams, child packages, elements. Elements are further arranged in order of type.

Within a type, components are initially listed in alphabetical or numerical order. You can change the order in which items are listed within their groups.

Use To:

- Adjust the order in which items are listed in the Project Browser

How To:

Step	Action	See also
1	<p>Click on an item in the Project Browser, then click on  or  in the toolbar at the top of the window</p> <p>The selected item is moved up or down in the Project Browser accordingly, but remains within its particular group of items</p>	

Step	Action	See also
2	To revert to listing components in alphabetical order, in the Project Browser right-click on the package and from the context menu select Contents Reset Sort Order	

2.3.8.2 Move Elements Between Packages

When creating the various diagrams, elements and packages that make up your model, at some point you might create an item under the wrong parent package.

In Enterprise Architect it is quite simple to relocate a model item from one parent package to another. Items can be relocated to a higher level package, a lower level package or a different package at the same level.

Use To:

- Move a diagram from one package to another
- Move an element from one package to another
- Move a package from one parent package to another

How To:

Step	Action	See also
1	In the Project Browser, click and drag the model item to be relocated from its existing position, and drop it onto the new parent package The dragged item and all of its child items are relocated to the target package	

Notes:

- Moving elements in the Project Browser does not affect the use of elements in diagrams
- Moving an element or package has no effect on any relationships that the element, package, or elements within the package have
- Moving a diagram generally does not affect the location of elements in packages
- However, elements of certain types, such as Initial Node, Decision and Final Node, are used only within one diagram and have no meaning outside that diagram; if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Learn More:

- [Initial Node](#)^[903]
- [Decision](#)^[886]
- [Final Node](#)^[895]

2.3.8.3 Move Elements in a Diagram

To improve the understanding of a diagram or to simply improve the presentation, you might want to adjust the position of the elements on your diagram.

Use To:

- Adjust the position of the elements on your diagram

How To:

For coarse adjustments:

Step	Action	See also
1	Click on a diagram element and drag it to its new position	

For fine adjustments:

Step	Action	See also
1	Click on a diagram element to select it	
2	While holding down the (Shift) key, press the arrow keys to move the selected element one pixel at a time in the direction of the arrow (Shift + ↑, ↓, →, ←)	

2.3.8.4 Move Elements Between Diagrams

If an element is in the wrong diagram, Enterprise Architect offers functionality that simplifies the process of moving the element from one diagram to another.

Use To:

- Move elements from one diagram to another
- Copy elements from one diagram to another

How To:

To move elements from one diagram to another, follow the steps below:

Step	Action	See also
1	In the Project Browser, double-click the name of the diagram that contains the element to be moved The diagram opens in the Diagram View	
2	Click on the element in the diagram to select it	
3	Either: <ul style="list-style-type: none"> • Click on the Cut button () in the Default Tools toolbar, or • Press (Ctrl + X) The selected element is copied to the clipboard However, the element remains visible on the current diagram until the clipboard content is pasted onto a different diagram The model element itself is not affected in any way	
4	In the Project Browser, double-click on the name of the diagram into which the element is to be moved The diagram opens in the Diagram View	

Step	Action	See also
5	<p>Either:</p> <ul style="list-style-type: none"> Click on the Paste button () in the Default Tools toolbar, or Press (Ctrl + V) or (Shift + Insert) <p>The element is pasted from the clipboard onto the current diagram and removed from the original (source) diagram</p> <p>Again, the model element itself is not affected in any way</p>	

To copy elements from one diagram to another, follow the steps below:

Step	Action	See also
1	<p>In the Project Browser, double-click the name of the diagram that contains the element to be moved</p> <p>The diagram opens in the Diagram View</p>	
2	Click on the element in the diagram to select it	
3	<p>Either:</p> <ul style="list-style-type: none"> Click on the Copy button () in the Default Tools toolbar, or Press (Ctrl + C) <p>The selected element is copied to the clipboard</p> <p>Neither the current diagram, nor the model element itself are affected in any way</p>	
4	<p>In the Project Browser, double-click on the name of the diagram into which the element is to be copied</p> <p>The diagram opens in the Diagram View</p>	
5	<p>Either:</p> <ul style="list-style-type: none"> Click on the Paste button () in the Default Tools toolbar, or Press (Ctrl + V) or (Shift + Insert) <p>The element is pasted from the clipboard onto the current diagram</p> <p>Again, neither the source diagram nor the model element itself are affected in any way</p>	

Notes:

- By selecting more than one element at a time, you can move or copy multiple elements in the same operation
- You can remove an element from a diagram by selecting it, then pressing the **Delete** key
- You can place an element onto an open diagram by locating the element in the Project Browser, then dragging it onto the diagram

2.3.8.5 Connect a Different Element on a Diagram

When creating connectors in your diagrams, at some point you might connect the wrong pair of elements. Enterprise Architect provides a simple method to enable you to move the end of a connector from one element to another.

Use To:

- Change the source element of a connector from the current element to another
- Change the target element of a connector from the current element to another

How To:

Step	Action	See also
1	In the diagram, click on the connector to select it The connector is highlighted with 'handles' at each connector end	
2	Move the cursor over the 'handle' at the end of the connector to reconnect The cursor changes to a small arrow () pointing upwards to the right	
3	Click and drag the connector end to the other element to connect to When you release the mouse button, the connector breaks from the original element and reconnects to the new element	

Notes:

- You can also tidy up a connection by dragging the end of the connector to a better position on the edge of the element, or move both ends at once by dragging the middle of the connector

2.3.9 Delete Elements

You can delete the elements of a model from a diagram or from the Project Browser.

Topic	Link
Remove elements from a diagram: this is essentially 'hiding' the element in that particular view of the model	Remove Elements from a Diagram ^[49]
Delete elements from a model - the element, its properties and any child elements or diagrams it has are all deleted; the element is removed from all diagrams on which it appears	Delete Elements from your Model ^[49]

Learn More:

- [Delete Connectors](#) ^[51]

2.3.9.1 Remove Elements From a Diagram

Removing an element from a diagram does not delete the element from the model and therefore does not alter the structure of the underlying model. When you remove an element from a diagram, you are essentially 'hiding' the element in that particular view of the model.

Use To:

- Remove one or more elements from a diagram

How To:

To remove elements from a diagram, follow the steps below:

Step	Action	See Also
1	Open the diagram that is to be modified Either: <ul style="list-style-type: none"> • Double-click on the diagram name in the Project Browser, or • Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View 	
2	Either: <ul style="list-style-type: none"> • Press (Ctrl+A) to select all of the elements in the diagram, or • Click on an element in the diagram to select it (and use (Ctrl + Click) to add further elements to your selection, or to remove elements from the selection) The selected elements are highlighted in the diagram	
3	Either: <ul style="list-style-type: none"> • Press (Delete), or • Press (Ctrl + D), or • Right-click a selected element and from the context menu select Delete <element name> or Delete Selected Elements. The selected elements are removed from the diagram.	

Notes:

- You can 'undo' the removal of a diagram element, by pressing (**Ctrl + Z**), or by clicking the **Undo** button , in the Default Tools toolbar

2.3.9.2 Delete Elements From Your Model

You can delete elements from your model, working either from the Project Browser or from a diagram.

When you delete an element from your model, the element, its properties and any child elements or diagrams it contains are all deleted. The deleted element is removed from all diagrams on which it appears and all relationships that it was involved in are also deleted from the model.

Use To:

- Delete one or more packages, elements or diagrams from your model

How To:

To delete one or more items from your model using the Project Browser, follow the steps below:

Step	Action	See Also
1	In the Project Browser, click on an item to select it To select additional items, or to remove items from the selection, use (Ctrl + Click) or (Shift + Click) The selected items are highlighted in the Project Browser	
2	Either: <ul style="list-style-type: none"> • Press (Ctrl + Delete) or • Right-click on one of the selected elements, then select the Delete selected item(s) context menu option A confirmation dialog is displayed	
3	Click on the Yes button The selected items are deleted from your model	

To delete one or more items from your model using a diagram, follow the steps below:

Step	Action	See Also
1	To open the diagram containing the items to be deleted from your model, either: <ul style="list-style-type: none"> • Double-click on the diagram name in the Project Browser, or • Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View 	
2	Either: <ul style="list-style-type: none"> • Press (Ctrl+A) to select all of the elements in the diagram, or • Click on an element in the diagram to select it, then use (Ctrl + Click) to add or remove specific elements to and from the selection The selected elements are highlighted in the diagram	
3	Press (Ctrl+Delete) A confirmation dialog is displayed	
4	Click on the Yes button The selected elements are removed from the model	

Notes:

- Deleting a package completely removes the package and all its contents - diagrams, child packages and elements - from the model
- Deleting an element completely removes the element and its properties, connectors, child elements and child diagrams from the model, and from every diagram that contains it
- Deleting a diagram completely removes the diagram from the model, but not the diagram's component elements; they remain in the parent package

2.3.10 Delete Connectors

It is possible to hide a connector on a diagram, or to delete it entirely from the model, removing the relationship between the previously connected elements.

Displaying all of the connectors on a complex diagram can make the diagram somewhat confused, so it can be useful to hide some of the connectors, to clarify a specific aspect of a more complex picture.

Use To:

- Hide a connector on a diagram
- Delete a relationship between two elements in your model

How To:

To hide or remove a connector between two elements, follow the steps below:

Step	Action	See also
1	Open the diagram containing the connector of interest Either: <ul style="list-style-type: none"> • Double-click the diagram name in the Project Browser, or • Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View 	
2	Click on the connector in the diagram to select it The selected connector is highlighted in the diagram	
3	Either: <ul style="list-style-type: none"> • Press (Ctrl + Delete) • Right-click on the connector and, from the context menu, select Delete Connector • Press (Delete) If you press (Ctrl + Delete), a dialog displays prompting you to confirm that you want to delete the connector (and the relationship it describes) from the model Otherwise, the Remove Connector dialog displays	
4	On the Remove Connector dialog, choose one of the options: <ul style="list-style-type: none"> • Hide the connector • Delete the connector from the model Hide the connector obscures the connector on the current diagram; you cannot see it but the relationship between the two elements still exists and is shown on other diagrams containing the two elements together, and on all reports of connectors between the elements Delete the Connector from Model removes the relationship that exists between the two elements; the connector is removed from the current diagram, from all other diagrams on which it is shown, and from all reports on connectors between the two elements	Hiding the Connector ^[75] Relationships ^[504]
5	Optionally, tick the checkbox Don't ask again Selecting this option prevents the Remove Connector dialog from being displayed next time you select the Delete Connector command; the command uses the setting you last used on the dialog	

Step	Action	See also
	Make sure that you have selected the right option to use as your default	
6	<p>Click on the OK button</p> <p>The connector disappears from the diagram</p> <p>If you selected Delete the connector from the model, the relationship represented by the connector is also deleted from your model</p>	

Notes:

- You cannot select multiple connectors - it is only possible to select one connector at a time
- If you select the **Don't ask again** checkbox on the Remove Connector dialog, you can reset this option on the Links page of the Options dialog (**Tools | Options | Links**)
- Selecting the **Hide the connector** option in the Remove Connector dialog, has the same effect as hiding the connector on the Links tab of the source element Properties dialog, or using the **Visibility | Hide Connector** context menu option

Learn More:

- To identify and reveal hidden connectors, see the [Links](#) ^[666] topic

2.3.11 Save Changes

Throughout much of your work in Enterprise Architect, any changes you make are automatically saved when you close the *dialog* (data entry window) on which you made the changes. In some cases the dialog contains a **Save** or **Apply** button, which enables you to save your changes and then keep working on the dialog.

If there is no specific dialog, such as when you create a diagram, you can save your work by:

- Clicking on the **Save** icon in the Diagram toolbar ()
- Pressing the (**Ctrl + S**) keyboard keys, or
- Selecting the **Diagram | Save** menu option

Often, Enterprise Architect does not let you close a screen without confirming that you want to save or discard your changes.

You can also save your diagram changes automatically, by selecting the **Auto Save Changes** checkbox on the Diagram Behavior page of the Options dialog.

Learn More:

- So far you have been using the Project Browser and Diagram View to develop your project; at this point you should find out a bit more about the other facilities of the Enterprise Architect [User Interface](#) ^[69] or Application Workspace
- When you have finished exploring the User Interface topics, go to the [Summary of Typical Tasks](#) ^[57] to identify areas of Enterprise Architect that provide particular support for your job role
- Explore other options for managing the way your diagrams display and function, on the [Diagram Behavior](#) ^[430] page

2.3.12 Create Requirements

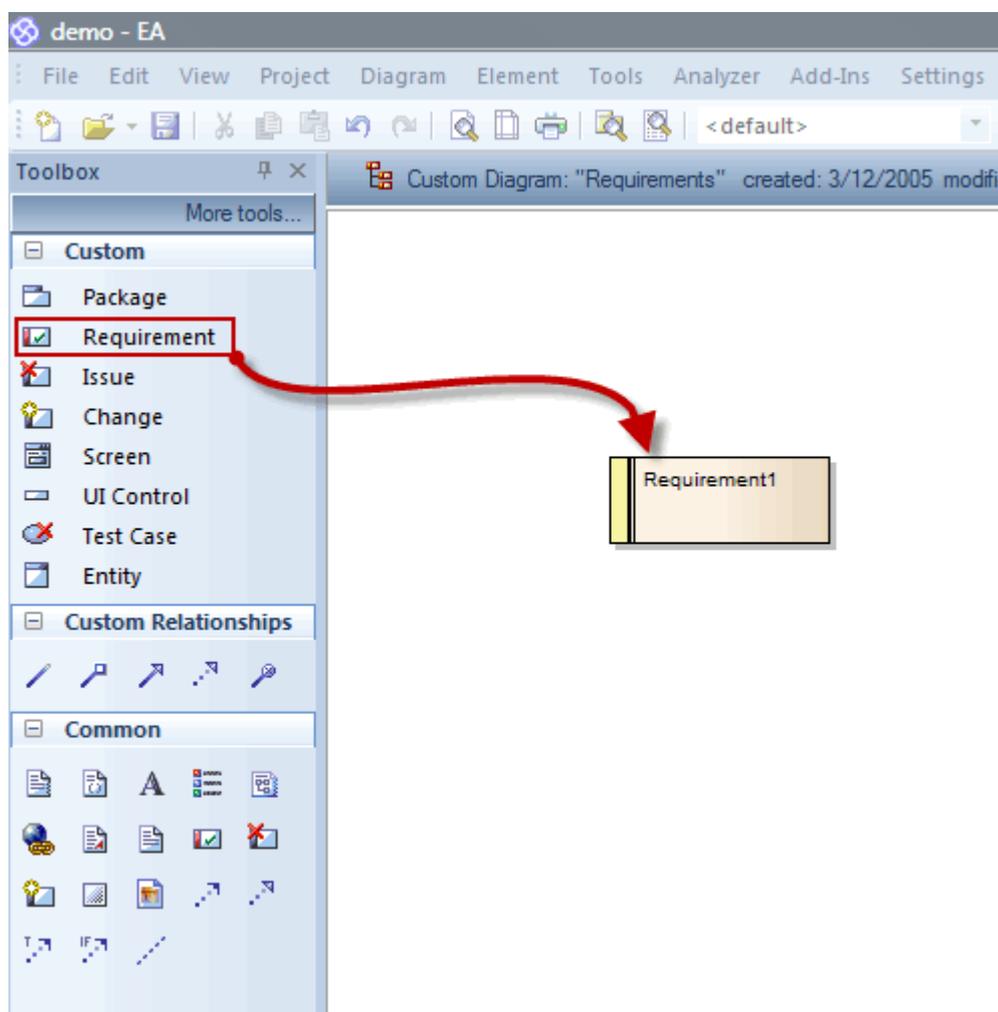
The following example illustrates how to create a requirement, name that element and provide appropriate notes.

Use To:

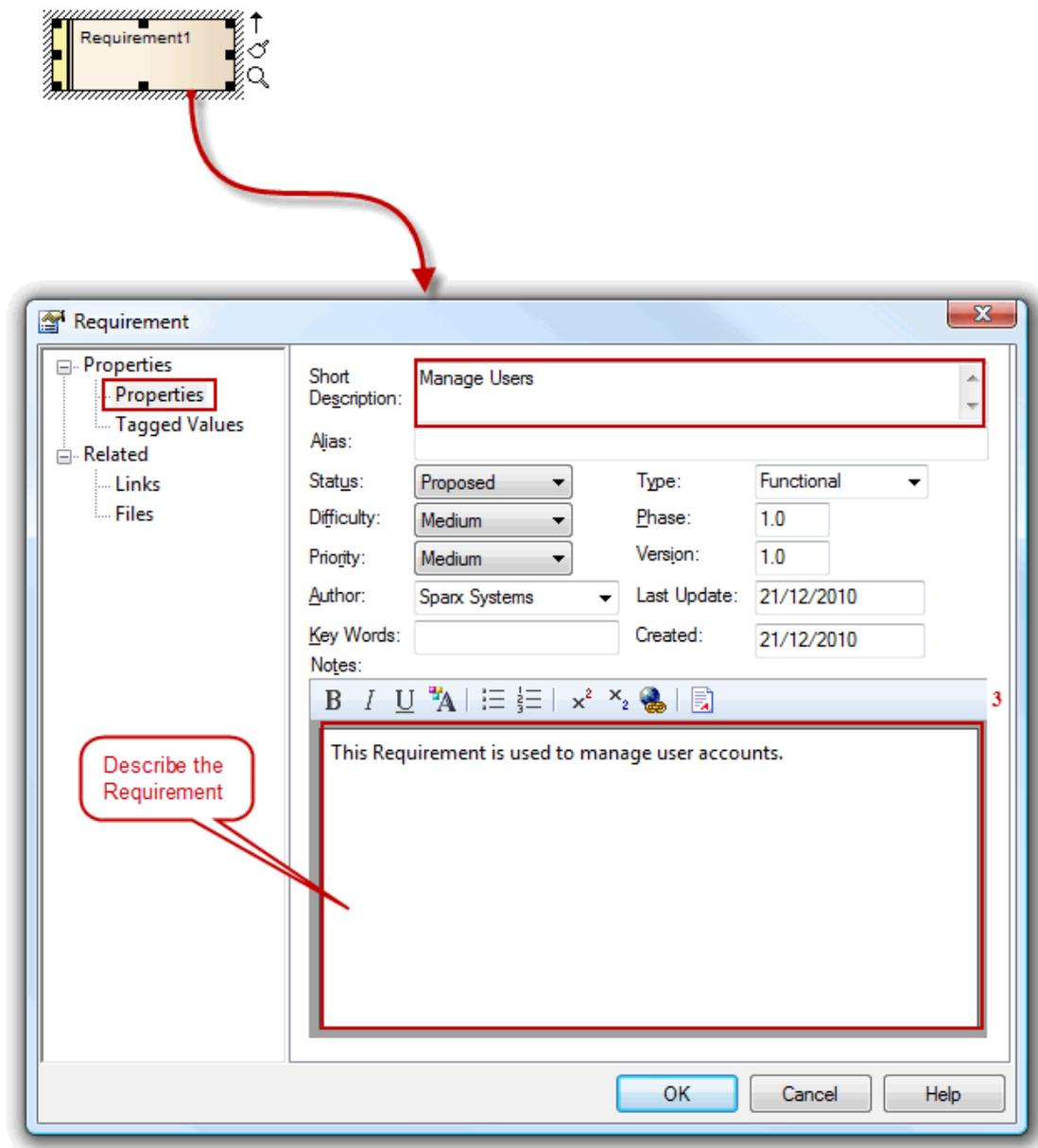
- Create a new Requirement element on a Requirements diagram.

How To:

The following images illustrate how to create a requirement, name that element and provide appropriate notes.

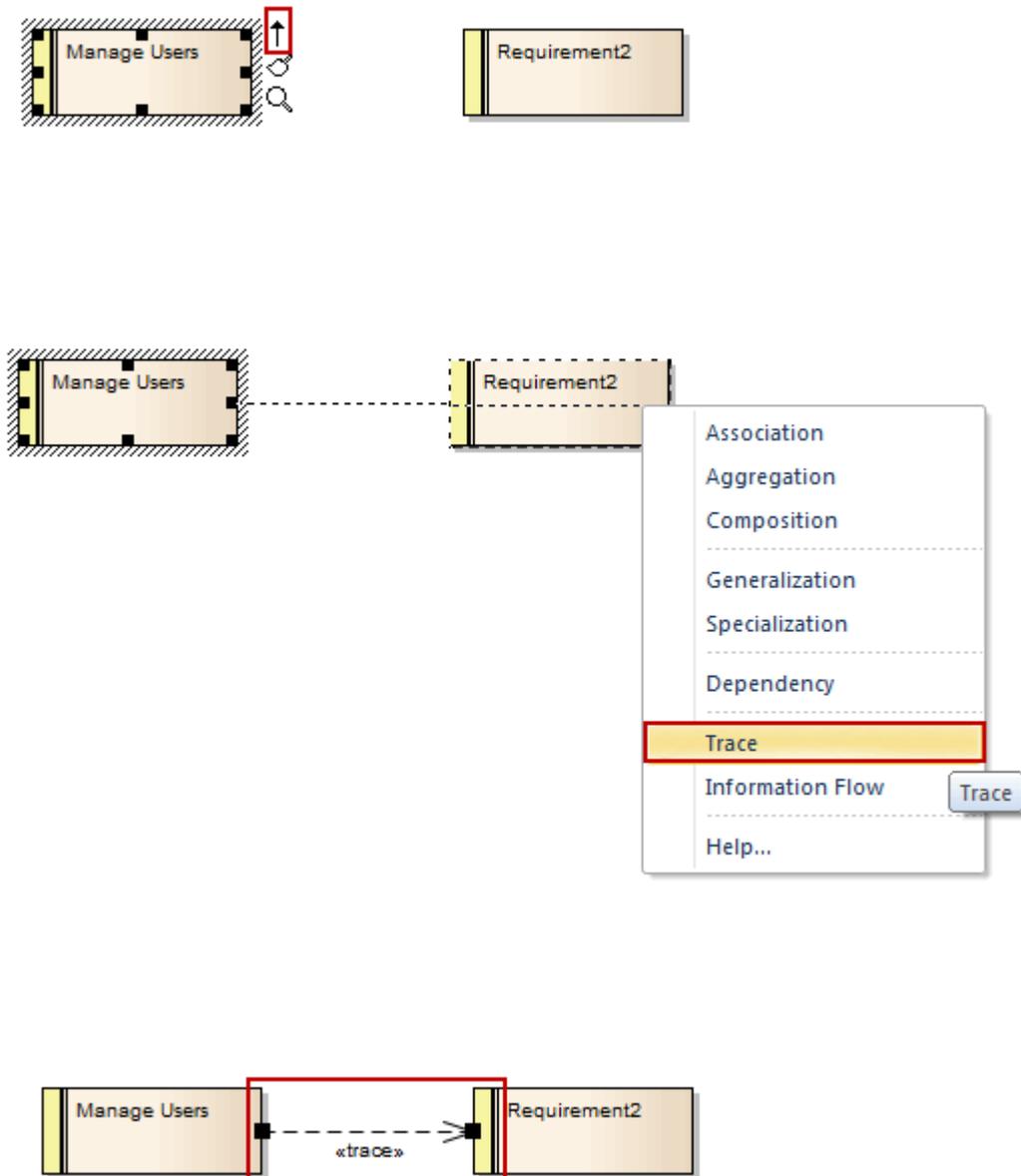


To access the Properties dialog, press **(Alt + Enter)**.

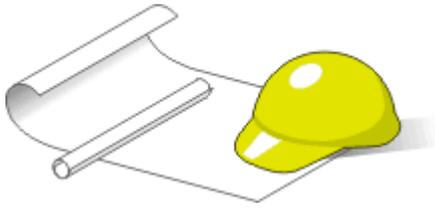


2.3.13 Requirement Trace Connector

Use the Quick Linker to create an appropriate connector between two Requirement elements.



2.4 Typical Project Roles



Enterprise Architect performs a number of tasks that are suited to a variety of professions. This topic describes some common working practices with Enterprise Architect for a range of project roles. There are tools for the roles of:

Topic	Link
Business Analyst, to create high-level models of business processes	Business Analyst ^[58]
Software Architect, to map functional requirements, perform real time modeling of objects, design the Deployment model and detail the deliverable components	Software Architect ^[59]
Software Engineer, to map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment and define software packages	Software Engineer ^[60]
Developer, to perform round trip code engineering, including reverse engineering of existing code and generation of code from Class elements	Developer ^[61]
Project Manager, to assign resources to elements, measure risk and effort, estimate project sizes, and manage element status, change control and maintenance	Project Manager ^[63]
Tester, to create test scripts against elements in the modeling environment	Tester ^[64]
Implementation Manager, to track and assign maintenance-related items to elements within Enterprise Architect, to rapidly capture and keep records of maintenance tasks such as issues, changes, defects and tasks, and to trace the maintenance of the items and processes involved in system deployment	Implementation Manager ^[65]
Technology Developer, to create customized additions to the functionality already present within Enterprise Architect	Technology Developer ^[65]
Database Developer, to develop databases, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model	Database Developer ^[67]

You can review a summary of the **typical tasks** supported for each role, or click on the appropriate role title to explore how Enterprise Architect can assist you in carrying out that role within a model driven project.

Learn More:

- [Summary of Typical Tasks](#)^[57]

2.4.1 Summary of Typical Tasks

Throughout a design and development project there are many different tasks to be performed, which could be carried out either by one person or - more probably - by members of a team with different responsibilities. In either case, Enterprise Architect supports most - if not all - of the responsibilities you might have on your project.

Therefore, the next topics to explore depend on the work you normally do on a project.

The descriptions below cover a number of job roles that Enterprise Architect supports. For those that most resemble your role on a project, follow the job title hyperlink to display a description of how that role might use Enterprise Architect, then follow links within those topics to explore some of the Enterprise Architect features of importance to the role.

Job Roles:

Role	Responsibilities	See also
Business Analyst	For modeling: <ul style="list-style-type: none"> • Requirements • High-level business processes • Business activities • Work flows • System behavior 	Business Analyst ^[58]
Database Developer	<ul style="list-style-type: none"> • Developing databases • Modeling database structures • Creating logical data models • Generating schema • Reverse engineering databases 	Database Developer ^[67]
Software Architect	<ul style="list-style-type: none"> • Mapping functional requirements of the system • Mapping objects in real time • Mapping the deployment of objects • Defining deliverable components 	Software Architect ^[59]
Tester	<ul style="list-style-type: none"> • Developing test cases • Importing requirements, constraints and scenarios • Creating Quality Test documentation • Tracking element defects and changes 	Tester ^[64]
Software Engineer	<ul style="list-style-type: none"> • Mapping Use Cases into detailed Classes • Defining the interaction between Classes • Defining system deployment • Defining software packages and the software architecture 	Software Engineer ^[60]
Project Manager	<ul style="list-style-type: none"> • Providing project estimates • Resource Management • Risk Management • Maintenance Management 	Project Manager ^[63]
Developer	<ul style="list-style-type: none"> • Forward, reverse and round-trip engineering • Visualizing the system states • Visualizing package arrangements • Mapping the flow of code 	Developer ^[61]

Role	Responsibilities	See also
Implementation Manager	<ul style="list-style-type: none"> Modeling the tasks in rolling-out a project, including network and hardware deployment Assigning and tracking maintenance items on elements (issues, changes, defects and tasks) 	Implementation Manager [65]
Technology Developer	For creating or customizing: <ul style="list-style-type: none"> UML Profiles UML Patterns Code Templates Tagged Value Types MDG Technologies Add-Ins 	Technology Developer [65]

Most of these roles work with specific types of diagram, so you might want to learn more about diagram types in general and specific types of diagram in particular.

Several types of project team member might want to generate documentation on their work and reports on how the project is developing and changing. Enterprise Architect enables you to generate project documentation in either RTF or HTML format.

Notes:

- The Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect have a user security feature that can be applied or turned off. If security is turned on, you require the appropriate access permissions to use many of the Enterprise Architect facilities listed above

Learn More:

- [List of Available Permissions](#) [205]
- [UML Diagrams](#) [795]
- [Report Generation](#) [1724]

2.4.2 Business Analysts

A Business Analyst can use Enterprise Architect to create high-level models of business processes. These include business requirements, activities, work flow, and the display of system behavior.

Using Enterprise Architect, a Business Analyst can describe the procedures that govern what a particular business does. Such a model is intended to deliver a high-level overview of a proposed system.

Topics:

Topic	Detail	See also
Model High Level Business Processes	With Enterprise Architect the Business Analyst can model high level processes of the business with Analysis diagrams Analysis diagrams are a subset of UML 2.3 Activity diagrams and are less formal than other diagram types, but they provide a useful means for expressing essential business characteristics and requirements	Analysis Diagram [1187] Example Analysis Diagram [1189]

Topic	Detail	See also
Model Requirements	<p>Gathering requirements is typically the first step in developing a solution, be it for developing a software application or for detailing a business process; it is an important step in the implementation of a project</p> <p>Enterprise Architect enables you to define the Requirement elements, connect Requirements to the model elements for implementation, connect Requirements together into a hierarchy, report on Requirements, and move Requirements out of model element responsibilities</p>	Requirements Management ^[1152]
Model Business Activities	<p>The Business Analyst can use Activity diagrams to model the behavior of a system and the way in which these behaviors are related to the overall flow of the system</p> <p>Activity diagrams do not model the exact internal behavior of the system but show instead the general processes and pathways at a high level</p>	Activity Diagram ^[814]
Model Work Flow	To visualize the cooperation between elements involved in the work flow, the Business Analyst can use an Interaction Overview diagram, which provides an overview of sub activities that are involved in a system	Interaction Overview Diagram ^[864]
Display System Behavior	In displaying the behavior of a system as a Use Case diagram, Enterprise Architect gives the Business Analyst an easily understood tool for mapping the functional requirements and behavior of a system	Use Case Diagram ^[813]

Learn More:

- [Business Modeling](#)^[1193]

2.4.3 Software Architects

Software Architects can use Enterprise Architect to map functional requirements with Use Cases, perform real time modeling of objects using *Interaction diagrams* (Sequence, Timing, Communication or Interaction Overview), design the Deployment model and detail the deliverable components using Component diagrams.

Topics:

Topic	Detail	See also
Map Functional Requirements of the System	<p>With Enterprise Architect the Software Architect can take the high level business processes that have been modeled by the Business Analyst and create detailed Use Cases</p> <p>Use Cases describe the proposed functionality of a system and are only used to detail a single unit of discrete work</p>	Use Cases ^[935]
Map Objects in Real Time	The Software Architect can use Interaction	Sequence Diagram ^[849]

Topic	Detail	See also
	<p>diagrams (Sequence and Communication diagrams) to model the dynamic design of the system</p> <p>Sequence diagrams are used to detail the messages that are passed between objects and the lifetimes of the objects</p> <p>Communication diagrams are similar to Sequence diagrams, but are used instead to display the way in which the object interacts with other objects</p>	<p>Example Sequence Diagram ^[85]</p> <p>Timing Diagram ^[83]</p> <p>Communication Diagram ^[85]</p> <p>Interaction Overview Diagram ^[86]</p>
Map Deployment of Objects	<p>The Software Architect can use Deployment diagrams to provide a static view of the run-time configuration of processing nodes and the components that run on the nodes</p> <p>Deployment diagrams can be used to show the connections between hardware, software and any middleware that is used on a system</p>	Deployment Diagram ^[80]
Detail Deliverable Components	<p>Component diagrams enable the Software Architect to model the physical aspects of a system</p> <p>Components can be executables, libraries, data files or another physical resource that is part of a system</p> <p>The component model can be developed from scratch from the Class model or can be brought in from existing projects and from third-party vendors.</p>	Component Diagram ^[80]

Learn More:

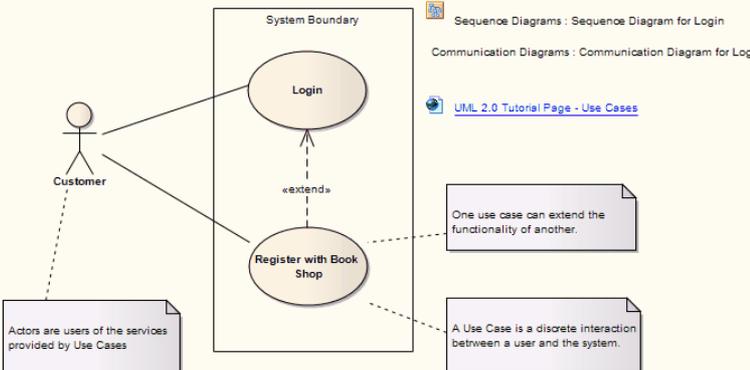
- [Analysis Diagrams](#) ^[118]
- [Modeling Fundamentals](#) ^[51]
- [XML Import and Export](#) ^[31]
- [XML Technologies](#) ^[158]

2.4.4 Software Engineers

Software Engineers using Enterprise Architect can map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment with Deployment diagrams and define software packages with Package diagrams.

Topics:

Topic	Detail	See also
Map Use Cases into Detailed Classes	<p>In Enterprise Architect the Software Engineer can study the Use Cases developed by the Software Architect, and with that information create Classes that fulfill the objectives defined in the Use Cases</p> <p>A Class is one of the standard UML constructs that is used to detail the pattern from which objects are produced at run time; to record the</p>	<p>Use Cases ^[81]</p> <p>Example Use Case Diagram ^[81]</p>

Topic	Detail	See also
	<p>relationships between Use Cases and Classes, the Software Engineer can create diagrams linking the elements with Realize connectors, and/or map the Realize connectors in the Relationship Matrix</p> 	<p>Class Diagram^[798]</p> <p>Realization^[1007]</p> <p>Relationship Matrix^[496]</p>
<p>Detail Interaction Between Classes</p>	<p>Interaction diagrams (Sequence and Communication diagrams) enable the Software Engineer to model the dynamic design of the system</p> <p>Sequence diagrams are used to detail the messages passed between objects and the lifetimes of the objects</p> <p>Communication diagrams are similar to Sequence diagrams, but are used instead to display the way in which objects interact with other objects</p>	<p>Sequence^[849]</p> <p>Communication^[859]</p>
<p>Define System Deployment</p>	<p>Deployment diagrams can be used to provide a static view of the run-time configuration of processing nodes and the components that run on the nodes</p> <p>Deployment diagrams can be used to show the connections between hardware, software and any middleware that is used on a system, to explain the connections and relationships of the components</p>	<p>Deployment^[804]</p>
<p>Define Software Packages</p>	<p>The Software Engineer can use Package diagrams to detail the software architecture</p> <p>Package diagrams are used to organize diagrams and elements into manageable groups, declaring the dependencies</p>	<p>Package^[796]</p>

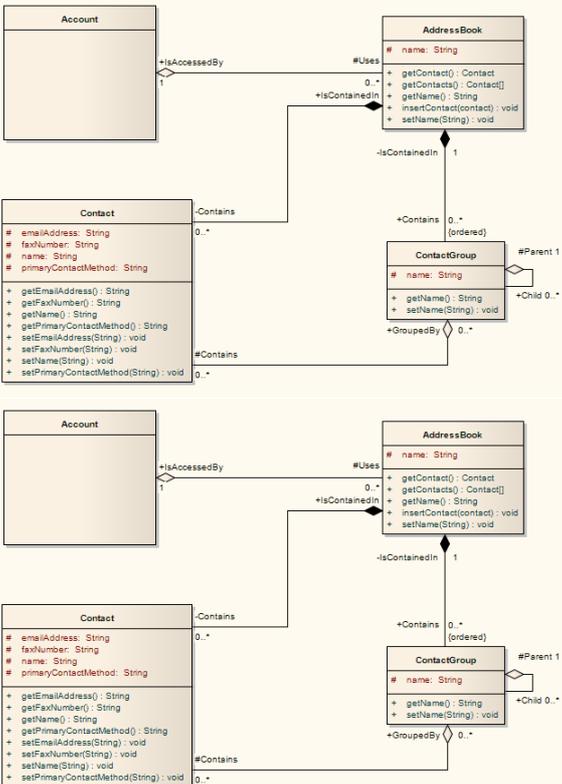
Learn More:

- [Modeling Fundamentals](#)^[515]

2.4.5 Developers

Developers can use Enterprise Architect to perform round trip code engineering, which includes reverse engineering of existing code and generation of code from Class elements.

Topics:

Topic	Detail	See also
Round Trip Engineering	<p>Enterprise Architect gives the developer unparalleled flexibility, with the ability to round trip software from existing source code to UML 2.3 models and back again</p> <p>Round trip engineering involves both forward and reverse engineering of code; keeping the model and code synchronized is an important aspect of this</p>	Code Engineering ^[1386] Synchronization ^[1506] Estimation ^[400]
Reverse Engineering	<p>Enterprise Architect enables developers to reverse engineer code from a number of supported languages and view the existing code as Class diagrams, which illustrate the static design view of the system</p> <p>Class diagrams consist of Classes and interfaces and the relationships between them; the Classes defined in UML Class diagrams can have direct counterparts in the implementation of a programming language</p>  <p>The diagram shows four classes: Account, AddressBook, Contact, and ContactGroup. Account has a relationship with AddressBook labeled '#IsAccessedBy' with multiplicity '1'. AddressBook has a relationship with Contact labeled '#Uses' with multiplicity '0..*' and another with Contact labeled '#IsContainedIn' with multiplicity '0..*'. AddressBook also has a relationship with ContactGroup labeled '#IsContainedIn' with multiplicity '1'. Contact has a relationship with ContactGroup labeled '#Contains' with multiplicity '0..*' and another with ContactGroup labeled '#Contains' with multiplicity '0..*'. ContactGroup has a relationship with ContactGroup labeled '#GroupedBy' with multiplicity '0..*'. ContactGroup also has a relationship with ContactGroup labeled '#Parent' with multiplicity '1' and another with ContactGroup labeled '#Child' with multiplicity '0..*'. The classes have various attributes and methods as shown in the diagram.</p>	Reverse Engineering ^[1506] Class Diagram ^[798] Class Element ^[941] Resources ^[348]
Forward Engineering	<p>As well as the ability to reverse engineer code, Enterprise Architect offers the developer the option of forward engineering code (code generation)</p> <p>This enables the developer to make changes to their model with Enterprise Architect and have these changes implemented in the source code</p>	Code Engineering ^[1386] Risk ^[351] Risk Types ^[355]
Determine the System State	<p>To visualize the state of the system the developer can use State Machine diagrams to describe how elements move between states, classifying their behavior</p>	State Machine ^[815]

Topic	Detail	See also
	<p>according to transition triggers and constraining guards</p> <p>State Machine diagrams are used to capture system changes over time, typically being associated with particular Classes (often a Class can have one or more State Machine diagrams used to fully describe its potential states)</p>	
Visualize Package Arrangement	Package diagrams are used to help design the architecture of the system; they are used to organize diagrams and elements into manageable groups, and to declare their dependencies	Package ^[796]
Follow the Flow of Code	<p>Activity diagrams are used to enable a better understanding of the flow of code</p> <p>Activity diagrams illustrate the dynamic nature of the system, which enables modeling of the flow of control between Activities and represents the changes in state of the system</p>	Activity ^[811]

Notes:

- State Machine, Package and Activity diagrams can be used by the developer to better understand the interaction between code elements and the arrangement of the code

Learn More:

- [Model Transformations - MDA](#)^[1301]
- [Visual Execution Analysis](#)^[1635]

2.4.6 Project Managers

Enterprise Architect provides support for the management of projects. Project Managers can use Enterprise Architect to assign resources to elements, measure risk and effort, and estimate project sizes. Enterprise Architect also helps them manage element status, change control and maintenance.

Topics:

Topic	Detail	See also
Provide Project Estimates	With Enterprise Architect the Project Manager has access to a comprehensive project estimation tool that calculates effort from Use Case and Actor objects, coupled with project configurations defining the technical and environmental complexity of the work environment	Estimation ^[400]
Resource Management	Managing the allocation of resources in the design and development of system components is an important and difficult task; Enterprise Architect enables the Project Manager or Development Manager to assign resources directly to model elements and track progress over time	Resources ^[348]
Risk Management	The Project Management window can be used to assign Risk to an element within a project; the risk types enable the Project	Risk ^[351]

Topic	Detail	See also
	Manager to name the risk, define the type of risk, and give it a weighting	Risk Types ^[355]
Maintenance	<p>Enterprise Architect enables the Project Manager to track and assign maintenance-related items to elements within Enterprise Architect; this enables rapid capture and record keeping for items such as issues, changes, defects, and tasks</p> <p>A Project Manager can also create and maintain a project glossary of processes, procedures, terms and descriptions</p>	Issues ^[356] Changes ^[1719] Defects ^[1719] Tasks ^[356] Project Glossary ^[362]

Learn More:

- [Projects Management](#) ^[346]
- [Element Status](#) ^[411]
- [Maintenance](#) ^[1719]

2.4.7 Testers

Enterprise Architect provides support for design testing by enabling you to create test scripts against elements in the modeling environment.

Topics:

Topic	Detail	See also
Test Cases	<p>With Enterprise Architect, Quality Assurance personnel can set a series of tests for each model element</p> <p>The test types include Unit testing, Acceptance testing, System testing and Scenario testing</p>	
Import requirements, constraints and scenarios	<p>To help ensure that testing maintains integrity with the entire business process, Enterprise Architect enables the tester to import requirements, constraints and scenarios defined in earlier iterations of the development life cycle</p> <p>Requirements indicate contractual obligations that elements must perform within the model</p> <p>Constraints are conditions that must be met in order to pass the testing process; constraints can be:</p> <ul style="list-style-type: none"> • Pre-conditions (states which must be true before an event is processed) • Post Conditions (events that must occur after the event is processed) or • Invariant constraints (which must remain true through the duration of the event) <p>Scenarios are textual descriptions of an object's action over time and can be used to describe the way a test works</p>	Requirements ^[1152] Constraints ^[666]
Create quality test documentation	Enterprise Architect provides the facility to generate high quality test documentation in the industry-standard .RTF file format	
Element defect	Defect tracking enables you to allocate defect reports to any	Element

Topic	Detail	See also
changes	element within the Enterprise Architect model; this enables all who are involved in the project to quickly view the status of defects, to see which defects have to be addressed and which have been dealt with	Defects ^[1715]

Learn More:

- [Introduction to Testing](#) ^[1694]

2.4.8 Implementation Managers

Enterprise Architect provides support for the management of project implementation. Enterprise Architect enables you to track and assign maintenance-related items to elements within Enterprise Architect. This enables you to rapidly capture and keep records of maintenance tasks such as issues, changes, defects and tasks. By providing a centralized facility for each element involved in the deployment process Enterprise Architect offers a powerful solution for tracing the maintenance of the items and processes involved in system deployment.

Topics:

Topic	Detail	See also
Deployment Diagram	<p>Using Deployment diagrams in Enterprise Architect, you can model the roll out of a project, including network deployment and workstation deployment</p> <p>Users involved in project deployment can add maintenance tasks to the diagram elements</p> <p>Deployment diagrams provide a static view of the run-time configuration of nodes on the network or of workstations, and the components that run on the nodes or are used in the workstations</p>	<p>Deployment Diagram ^[804]</p> <p>Example Deployment Diagram ^[806]</p>

Learn More:

- [Project Maintenance](#) ^[1713]

2.4.9 Technology Developers

Technology Developers are Enterprise Architect users who create customized additions to the functionality already present within Enterprise Architect.

Additions include UML Profiles, UML Patterns, Code Templates, Tagged Value Types, Scripts, Custom Queries, Transformations, MDG Technologies and Enterprise Architect Add-Ins. By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.

Topics:

Topic	Detail	See Also
-------	--------	----------

UML Profiles	<p>By creating UML Profiles the technology developer can create a customized extension for building UML models that are specific to a particular domain</p> <p>Profiles are stored as XML files and can be imported into any model as required</p>	UML Profiles ^[1026]
UML Patterns	<p>Patterns are sets of collaborating Objects and Classes that provide a generic template for repeatable solutions to modeling problems</p> <p>As patterns are discovered in any new project, the basic pattern template can be created</p> <p>Patterns can be re-used with the appropriate variable names modified for any future project</p>	Patterns ^[1021]
Code Templates	<p>Code templates are used to customize the output of source code generated by Enterprise Architect; this enables the generation of code languages not specifically supported by Enterprise Architect and enables you to define the way Enterprise Architect generates source code to comply with your own company style guidelines</p>	Code Templates ^[1482]
Tagged Value Types	<p><i>Tagged Values</i> are used in Enterprise Architect to extend the information relating to an element in addition to the information directly supported by the UML language</p> <p>A Tagged Value, strictly, is the value of a property of a modeling item, the property being called a tag; for example: a Class element called Person might have a tag called Age with the Tagged Value of 42</p> <p>More loosely, the combination of tag and value can be referred to as a Tagged Value</p> <p>A Tagged Value type is a group of parameters that define and/or limit the possible values of a tag and, in many instances, how a specific value is assigned to the tag; for example, the tag Age might have a Tagged Value Type of Integer, so the user simply types in a numeric value</p> <p>Alternatively, the type could be Spin, with lower and upper limits of, say, 20 and 120, so the user sets a value by clicking on arrows in the field to increment or decrement the value within the limits of 20 and 120</p> <p>Typically, Tagged Values are used during the code generation process, or by other tools to pass on information that is used to operate on elements in particular ways</p>	Tagged Values ^[762] Tagged Value Type ^[775]
MDG Technologies	<p>MDG Technologies can be used to create a logical collection of resources that can contain UML Profiles, Patterns, Code Templates, Image files and Tagged Value types that are accessed through a technology file</p>	MDG Technologies ^[1031]
Enterprise Architect Add-Ins	<p>Add-Ins enable you to build your own functionality into Enterprise Architect, creating your own mini programs that can extend the capabilities of Enterprise Architect, defining your own menus, and creating your own Custom Views</p>	Enterprise Architect Add-Ins ^[1967]

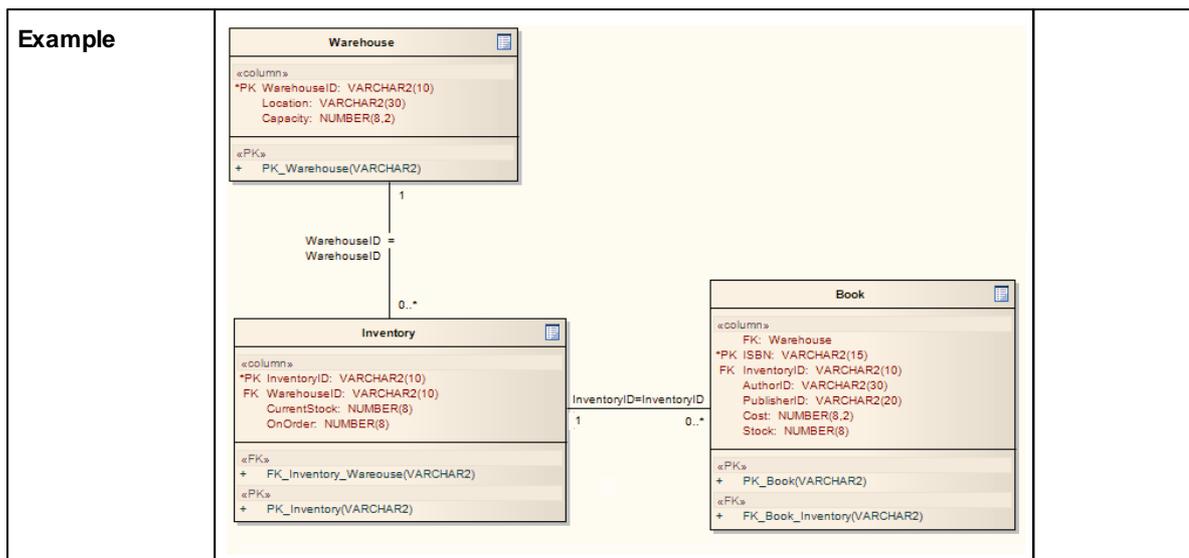
Learn More:

- [Extending UML](#) ^[1038]
- [MDG Technology SDK](#) ^[1038]

2.4.10 Database Developers

Enterprise Architect supports a range of features for the development of databases, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model.

Topic	Detail	See Also
Create Logical Data Models	<p>With Enterprise Architect the Database developer can build database diagrams using the built-in UML Data Modeling Profile</p> <p>This supports the definition of Primary and Foreign keys, cardinality, validation, triggers, constraints and indexes</p>	Logical Data Model ^[1279]
Generate Schema	<p>By using Enterprise Architect's DDL generation function the Database Administrator can create a DDL script for creation of the database table structure from the model</p> <p>Enterprise Architect currently supports:</p> <ul style="list-style-type: none"> • JET-based databases • DB2 • InterBase • Informix • Ingres • MySQL • SQL Server • PostgreSQL • Sybase Adaptive Server Anywhere and Adaptive Server Enterprise, and • Oracle 9i, 10g and 11g 	Schema ^[1278]
Reverse Engineer Database	<p>Using an ODBC data connection the Database Administrator can import a database structure from an existing database to create a model of the database</p> <p>Generating the model directly from the database enables the DBA to quickly document their work and create a diagrammatic account of a complex database through the graphical benefits of UML</p>	Reverse Engineering ^[1508]

**Learn More:**

- [Data Models](#) ^[1276]
- [Physical Data Model](#) ^[1346]

2.5 User Interface Guide



This section describes the *Enterprise Architect Application Workspace*, which provides a powerful and flexible analysis, modeling, design and construction environment. The workspace is built from familiar Windows components, such as toolbars, docking windows, dialogs and menus. Together these elements provide a **rich set of tools** for working with diagrams, models, **UML**, source code, executing applications (debugging and visualization), rich text documents, collaborative tools, extension technologies and much more.

Topics:

Topic	Link
The Enterprise Architect workspace, the interface through which you create and display your models; the main workspace component is the central Diagram View , which is where you create model diagrams, view reports, scroll through lists of model elements, access the internet and even edit and debug source code	The Start Page ^[70] Diagram View ^[540] The Web Browser ^[128]
The Main Menu , which provides access both to high-level functions related to the project life cycle, and to project and system administration functions; throughout Enterprise Architect you can also access functions and operations using context menus	The Main Menu ^[73] Project Browser Context Menus ^[443] Diagram Context Menu ^[538]
The windows that are used to enter and display information on your project and models; you can rearrange windows to adapt the screen space to your work habits	Standard Windows ^[102]
Toolbars are small sets of buttons that provide quick access to perform common operations both on your project as a whole and on the individual modeling units	Standard Toolbars ^[106]
The Enterprise Architect Diagram Toolbox provides all the components and connectors that you use to create models using whatever diagrams are appropriate; the Toolbox automatically matches the kind of diagram you have open, and whichever technology is currently active in your model	The Diagram Toolbox ^[546]
The key combinations that you can use throughout Enterprise Architect to quickly initiate a wide range of actions	Keyboard Shortcuts ^[128]
The customization tools that enable you to tailor the menus, toolbars and function keys that you use in Enterprise Architect, to set the visual style of the application, and to customize the workspace layout	Customization ^[118] Visual Styles ^[125] Manage Workspace Layout ^[126]
The navigation and search tools , which enable you to search for, track and locate information in your project, and filter the information so that you can explore particular views of your project rather than the entire contents	Navigate, Search and Trace ^[440] The Project Browser ^[441]

Learn More:

- [A Demonstration of Enterprise Architect in use](#)

2.5.1 Start Page

The Start Page provides some useful navigation tools and jumping off points when you first open Enterprise Architect - including the **recent files list**, commands to **create or copy projects**, connect to **repository servers** and useful **Help topics** to get you started. Links to online resources and to the Sparx Systems Enterprise Architect Community website can also help you get up to speed and explore the possibilities of Enterprise Architect.

If you have no projects of your own to open yet, open the *EAExample* project or use the *QuickStart Tutorial* to create a project to explore.

Actions:

Action	Usage	See Also
Search	<p>Locate an object in Enterprise Architect; type the name of the object in this field and click on the (...) button</p> <p>Enterprise Architect displays the results of the search on the Model Search screen</p> <p>Click on an item in the search results to highlight it in the Project Browser</p>	<p>Model Search^[475]</p> <p>Project Browser^[441]</p>
Open a Project File	<p>Display the Select Enterprise Architect Project to Open dialog, which you use to open an existing project (where you have more project files than can be listed in the Recent panel)</p>	<p>Open a Project^[138]</p>
Create a New Project	<p>Save a new project and open the Model Wizard dialog</p>	<p>Create a New Project^[145]</p> <p>Model Wizard^[518] dialog</p>
Copy a Base Project	<p>Select a different Base Project to generate a new project from</p>	<p>Copy a Base Project^[146]</p>
Connect to Server	<p>Specify a Data Source name to connect to; the supported repositories are listed on the right</p> <p>This feature is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions</p>	<p>Connect to Server^[161]</p> <p>MySQL^[149]</p> <p>SQL Server^[150]</p> <p>Oracle 9i, 10g and 11g^[151]</p> <p>PostgreSQL^[151]</p> <p>MSDE^[152]</p> <p>Adaptive Server Anywhere^[152]</p> <p>Access 2007^[149]</p> <p>Progress OpenEdge^[153]</p>
Getting Started	<p>Open the Learning Center, to display useful topics and guides for various areas of work in Enterprise Architect</p>	<p>Learning Center^[72]</p>

Action	Usage	See Also
Configure Options	Display the Help on the Options dialog, which enables you to define how Enterprise Architect displays and processes information.	Options ^[42] dialog
Online Resources & Tutorials	Open the Resources page of the Sparx Systems website, which provides access to a wide range of Enterprise Architect and UML tutorials, demonstrations, examples, Add-Ins and advice	The Web Browser ^[128]
Enterprise Architect Community	Access the Sparx Systems Enterprise Architect Community Site, which contains a range of articles, discussions, tools and resources provided by both Sparx Systems and the Enterprise Architect user community You must register to use the facilities of the site; you can also register as an author and submit material yourself, for others to read and use	Sparx Systems Enterprise Architect Community Site
Recent	Select from a list of hyperlinks to the most recently used Enterprise Architect projects (both .EAP files and DBMS connections) If necessary, you can remove the hyperlink to a project from this list	Remove Recent Projects ^[71]

Notes:

- If your model has a default diagram set, that diagram opens immediately, hiding the Start Page; you can still access the Start Page from the diagram tabs below the diagram
- If you have set a shortcut view profile, that overrides the default diagram setting
- You can also hide the Start Page, and show it again, by de-selecting and selecting the **Window | Show Start Page** option; the setting persists though shut down and restart, until you specifically change the option again

Learn More:

- [Quickstart Tutorial](#) ^[37]
- [Set the Default Diagram](#) ^[583]
- [Diagram Tabs](#) ^[543]

2.5.1.1 Remove Recent Projects

Removing the hyperlink to a project from the Start Page only removes the link to the project and does not remove the project from the file system or repository.

Access: **File | Open Project > Open Enterprise Architect Project > Recent Projects: Remove Selection from List (Ctrl+O)**

How to:

To remove a project hyperlink from the Recent list on the Start Page, follow the steps below:

Step	Action	See Also
1	On the Open Enterprise Architect Project dialog, in the Recent Projects panel, click on the project to remove	
2	Click on the Remove Selection from List button	

2.5.2 Learning Center

The Learning Center window provides quick access to a range of context-specific Enterprise Architect help topics, online resources and facilities in a number of specific work areas in Enterprise Architect, such as Getting Started, Modeling or Testing.

These topic areas can include topics specific to any MDG Technologies being used with Enterprise Architect.

Access: **View | Learning Center (Ctrl + Shift + 9)**

(Also, when you first open Enterprise Architect, the Learning Center automatically displays on the right of the screen)

Uses:

- Initiate a task or process within a specific work area
- Display topics from Enterprise Architect Help that describe either procedures or facilities to achieve objectives in a work area
- Display web pages or documents that provide conceptual information or examples within a work area
- Run selected reports or data collation tools
- Run demonstrations of functions within a work area

Actions:

To display information listed in the Learning Center, click on the required item. The types of item are indicated by the list titles and by the following icons:

Icon	Action
	Open appropriate topics from the Enterprise Architect Help file
	Open web pages or documents on the Sparx Systems web site
	Begin Enterprise Architect tasks appropriate to the Learning Center topic area; you must be in an appropriate functional area of Enterprise Architect in order for these tasks to function, such as in an open diagram
	Begin Add-In tasks appropriate to the Learning Center topic area; you must be in an appropriate functional area in order for these tasks to function
	Open report facilities to provide information or data collation tools
	Start demonstrations of Enterprise Architect functions in action

The selected information, web page or demonstration displays on a Browser tab in the main view, or the appropriate task or report window opens.

To switch between the topic areas, either:

- Click on the drop-down arrow at the right of the Learning Center toolbar and select the required area from the list, or
- Click on the left or right arrow buttons in the toolbar

The 'Home' icon returns you to the Getting Started topic area.

2.5.3 Main Menu

The Enterprise Architect **Main Menu**, at the top of the Enterprise Architect workspace, opens a range of subordinate menus that provide mouse-driven access to many high-level functions related to the project life cycle, as well as administration functions.

Each subordinate menu is described in the following table:

Usage	See Also
File - provides options to create, open, close and save projects, and also to perform print tasks	File ^[74]
Edit - provides a range of functions to apply to diagram elements for the currently open diagram or view	Edit ^[74]
View - enables you to display each of the windows or toolbars, and to set the visual style of your Enterprise Architect workspace	View ^[76]
Project - provides access to tasks related to the management of your project, such as adding to the project structure, importing and exporting data, generating documentation, version control and security	Project ^[80]
Diagram - enables you to configure diagram properties and options and to save diagram images to file	Diagram ^[83]
Element - provides options to configure and access element features and details and control element layout, and to generate documentation and work on source code for the element	Element ^[85]
Tools - provides access to tools related to code engineering, model management, spell checking, customization of features and setting operational options	Tools ^[90]
Analyzer - provides access to a range of functions to perform build, debug and profiling operations on your model code	Analyzer ^[94]
Add-Ins - enables you to connect to, display information on, work with and manage your Add-Ins	Add-Ins ^[97]
Settings - enables you to configure various settings for your overall project, such as stereotypes, Tagged Values, cardinality values, datatypes, language macros, local directories, image management	Settings ^[98]
Window - provides various functions for managing open system windows.	Window ^[100]
Help - provides access to the Enterprise Architect help files, the Read Me file, the Enterprise Architect End User License Agreement, the Enterprise Architect Example model and various features on the Sparx Systems website; it also enables you to manage your license keys	Help ^[101]

Additionally, if you right-click on the **Toolbar** area just under the menu bar, a composite context menu displays providing options to display the toolbars and all significant windows and views.

2.5.3.1 File

The **File** menu provides options to create, open, close and save projects, and also to perform print tasks.

Access: **File**

Option	Usage	Shortcut	See also
New Project	Create a new Enterprise Architect project	Ctrl + N	File Based Repositories ^[145]
Open Project	Open a project	Ctrl + O	Open a project ^[138]
Reload Current	Reload the current project (use in a multi-user environment to refresh the Project Browser)	Ctrl + Shift + F11	Refresh View of Shared Project ^[187]
Close Project	Close the current project		
Save Project Copy	Save the current model with a new name		Copy Existing Project ^[147]
Save Project Shortcut	Create a desktop shortcut to the current model (This option is also active in the 'Lite', read-only edition of Enterprise Architect)		Project Shortcuts ^[140] The Read-Only 'Lite' Edition ^[18]
Print Setup	Configure your printer's settings		
Page Setup	Configure the page settings to print the current diagram		Set Diagram Page Size ^[600]
Print Preview	Preview ^[540] how the currently displayed diagram prints		Print Preview ^[540]
Print	Print the currently displayed diagram Enterprise Architect provides facilities to change the scale of the printed diagram (the number of pages it takes up) and to print or omit page headers and footers on the diagram	Ctrl + P	Scale Image To Page Size ^[602] Diagram Tab ^[572]
Recent Files List	Select from a list of the most recently opened projects		
Exit	Exit Enterprise Architect		

2.5.3.2 Edit

The **Edit** menu provides a range of functions to apply to diagram elements for the currently open diagram or view.

Access: **Edit**

Option	Usage	Shortcut	See also
Model Search	Display the Model Search window, to search for particular phrases or words	Ctrl + Alt + A	Model Search ^[475]
File Search	Display the File Search window, to search for text in code files and scripts		Find In Files ^[145]

Option	Usage	Shortcut	See also
Copy	<ol style="list-style-type: none"> Copy the selected elements to the MS Windows clipboard. (To paste the selected elements use the Paste Element(s) option, below) Copy an image of the selected elements to the clipboard. If no elements are selected, the entire diagram is copied <p>The image can be saved as a bitmap or a metafile; you set the format on the Options dialog</p>	Ctrl + C	General ^[42] (page of the Options dialog)
Project Clipboard	<p>Offers two sub-options:</p> <ul style="list-style-type: none"> Add to Project clipboard - Add the current element to the Enterprise Architect clipboard Clear Project Clipboard - Clear any elements from the Enterprise Architect clipboard 	Ctrl + Space ---	
Paste Element (s)	Display the Paste Elements submenu, which enables you to paste elements on the clipboard into the current diagram.		Paste Elements Submenu ^[75]
Paste Image From Clipboard	<p>Paste the element in the Enterprise Architect Clipboard into the same diagram or a different diagram, as a metafile (that is, a definition of the element) as many times as is necessary</p> <p>If you paste the element into a different diagram, its classifier identifies the source diagram for the element</p>	Ctrl + Shift + Insert	
Redo	Re-apply the last undone action	Ctrl + Y	Redo Last Action ^[603]
Undo	Undo the last action performed; note that some actions cannot be undone	Ctrl + Z	Undo Last Action ^[603]
Select All	Select all elements concurrently on the current diagram		
Select By Type	Specify a particular type of element to select on the diagram		
Clear Selection	Deselect all elements		
Bookmark Selected	Bookmark the selected element(s). If the selected element is already bookmarked, this option removes the bookmark	Shift + Space	Manage Bookmarks ^[412]
Clear All Bookmarks	Clear bookmarks from any bookmarked elements in the current diagram		Manage Bookmarks ^[412]
Delete Selected Element(s)	Delete the selected elements from the diagram	Ctrl + D	

2.5.3.2.1 Paste Elements Submenu

You can paste an element from the Enterprise Architect Clipboard or the MS Windows clipboard into a diagram, either as a hyperlink to the original element or as a new element. The Enterprise Architect clipboard takes precedence, so you must clear that clipboard before you can paste from the MS Windows clipboard.

Access: **Edit | Paste Element(s)**

Option	Usage	Shortcut	See also
as Link	<p>Paste the element in the buffer as a link (that is, a reference) to the element</p> <p>If there are images in the MS Windows clipboard and none in the Enterprise Architect clipboard, you can:</p> <ul style="list-style-type: none"> • Paste an image from the MS Windows clipboard into a new element as the appearance of the new element, or • Paste an image from the MS Windows clipboard into the diagram as a new boundary's appearance 	Shift + Insert	Paste from Project Browser ^[57]
as New	Paste the element in the buffer as a completely new ^[56] element	Ctrl + Shift + V	

2.5.3.3 View

The **View** menu enables you to display and/or focus on each of the windows or toolbars, and to set the visual style of your Enterprise Architect workspace.

Access: **View**

Option	Usage	Shortcut	See also
Workspace Layouts	Show the Workspace Layout dialog for changing the content and layout of the Enterprise Architect workspace		Manage Workspace Layout ^[12]
Learning Center	Show the Learning Center, providing context-specific help topics, online resources and facilities in a number of specific work areas in Enterprise Architect	Ctrl + Shift + 9	Learning Center ^[72]
Project Browser	Display the Project Browser, for browsing and exploring your model; this displays the hierarchical arrangement of packages, elements and features within the model	Alt + 0	The Project Browser ^[44]
Package Browser	Display the elements of the current package in the Package Browser, as a context-sensitive, editable table	Ctrl + Alt + R	Package Browser ^[45]
Model Views	Show the Model Views window, through which you can create filtered views of selected areas or aspects of your model	Ctrl + Shift + 5	Model Views ^[46]
More Project Tools	Display a submenu containing options for the Resources and Project Information windows, and the internal Web Browser		View Submenus ^[78]
Personal Information	Display the Personal Information window, which enables you to email project team members		Personal Information ^[37]

Option	Usage	Shortcut	See also
	and report on the work and tasks assigned to you		
Team Review	Show the Team Review window, the portal for a discussion forum within your development team community	Ctrl + Alt + U	Team Review Tools ^[216]
Task Allocations	Show the Project Task Allocation window, through which you manage work tasks allocated within the project and run reports on the components associated with the tasks		Project Task Allocation ^[366]
Project Calendar	Display the Calendar, to monitor work across the project in terms of the resources involved, the tasks being performed, the events occurring and the elements being worked on		Project Calendar ^[390]
Diagram Toolbox	Show the Diagram Toolbox, a panel of icons for creating elements and connectors on a diagram; there are pages of icons for each UML, Extended and MDG Technology diagram type	Alt + 5	Diagram Toolbox ^[546]
Diagram Filters	Display the Diagram Filters window, for selecting the elements to show or hide on a diagram		Diagram Filters ^[489]
Diagram Layout	Display the Layout Tools window, for reformatting your diagram in one of a range of layouts		Layout Diagrams ^[604]
Pan and Zoom	Display the Pan & Zoom window for panning across a diagram to display sections at greater magnification	Ctrl + Shift + N	The Pan & Zoom Window ^[473]
Element Properties	Show the Properties window for the selected element, which provides an editable table of the common properties of the selected element	Alt + 1	Properties Window ^[689]
Relationships	List a selected element's connectors on the Relationships window	Ctrl + Shift + 2	The Relationships Window ^[504]
Traceability	Show the Traceability ^[495] window, for tracing the relationships of an element through the model	Ctrl + Shift + 4	The Traceability Window ^[495]
More Element Tools	Display a submenu containing options for the Scenarios & Requirements, Element Browser, Project Management, Maintenance, Testing and Source Code windows		View Submenus ^[78]
Relationship Matrix	Display the Relationship Matrix, to cross reference elements to each other by connector type		Relationship Matrix ^[496]
Gap Analysis Matrix	Display the Gap Analysis Matrix to analyze model artifacts for potential gaps in the solution		Gap Analysis Matrix ^[510]
Notes	Show the Notes window, to create and review the explanatory text on a diagram, element, feature or connector	Ctrl + Shift + 1	Notes ^[769]

Option	Usage	Shortcut	See also
Tagged Values	Show the Tagged Values window, to add further properties to a modeling object beyond those supported by UML	Ctrl + Shift + 6	Tagged Values ^[76]
System Output	Show the Output window	Ctrl + Shift + 8	The System Output Window ^[127]
Scripting	Show the Scripting window		Scripting ^[1817]
Toolbars	Show or hide individual toolbars		View Submenus ^[78]
Visual Style	Set the Visual Style and a number of other display options		View Submenus ^[79]

2.5.3.3.1 View Submenus

The More Project Tools Sub-Menu:

Select the windows to be visible and deselect those to be hidden. You can select from:

Option	Usage	Shortcut	See also
Project Resources	Show or hide the Resources window	Alt+6	Resources ^[789]
Project Information	Show or hide the Project Information window	Alt+2	The Project Information Window ^[356]
Internal Web Browser	Open the web browser page at the site you have specified on the Options dialog, in the Web Home field	Ctrl+Alt+W	Internal Web Browser ^[128] Options ^[422]

The More Element Tools Submenu:

Select the windows to be visible and deselect those to be hidden. You can select from:

Option	Usage	Shortcut	See also
Scenarios & Requirements	Display the Scenarios, internal Constraints and Requirements for the selected element, as tabs of the main work space.	Ctrl+Shift+3	Scenarios, Internal Constraints and Requirements ^[689]
Element Browser	Explore the components of the selected element, in the Element Browser window.	Alt+9	Element Browser ^[687]
Project Management	Show or hide the Project Management window	Ctrl+Shift+7	Project Management ^[347]
Maintenance	Show or hide the Maintenance window	Alt+4	Maintenance ^[1713]
Testing	Show or hide the Testing window	Alt+3	Testing ^[1695]
Source Code	Show or hide the Source Code window.	Alt+7	The Source Code Viewer ^[1408]

The Toolbars Sub-Menu:

Select the toolbars to be visible and deselect those to be hidden. You can select from:

Topic	Usage	Link
Default Tools	Show or hide the Default Tools toolbar	Default Tools ^[107]
Project	Show or hide the Project toolbar	Project ^[108]
Code Generation	Show or hide the Code Generation toolbar	Code Generation ^[109]
UML Elements	Show or hide the UML Elements toolbar	UML Elements ^[110]
Diagram	Show or hide the Diagram toolbar	Diagram ^[111]
Current Element	Show or hide the Current Element toolbar	Current Element ^[112]
Current Connector	Show or hide the Current Connector toolbar	Current Connector ^[114]
Workspace Layouts	Show or hide the Workspace Layouts toolbar	Workspace Layouts ^[113]
Debug & Record	Show or hide the Debug & Record toolbar	Debug & Record ^[116]
Status Bar	Show or hide the Status Bar	Status Bar ^[117]

The Visual Style Sub-Menu:

Option	Usage	Shortcut	See also
Select Visual Style	Select different visual styles or <i>themes</i> for the Enterprise Architect user interface		Visual Styles ^[125]
Animate Autohide Windows	Animate windows that have been automatically hidden		Automatically Hidden ^[105]
Show Toolbar Customize Button	Toggle the down-arrow on the end of each toolbar that enables you to customize the toolbar buttons, as shown below: 		
Hide Diagram Caption Bar	Hide or redisplay the diagram caption bar at the top or bottom of a diagram		Caption Bar ^[540]

2.5.3.4 Project

The **Project** menu gives access to tasks related to the management of your project, such as adding to the project structure, importing and exporting data, generating documentation, version control and security.

Access: **Project**

Action	Usage	Shortcut	See also
New Model	Display the Model Wizard, to add models to your project		Model Wizard ^[518]
New Package	Create a new package	Ctrl + W	Create a new package ^[534]
New Diagram	Create a new diagram in the current package	Ctrl + Insert	Create a new diagram ^[568]
Documentation	Display the Documentation Submenu		Documentation Submenu ^[80]
Model Validation	Display the Model Validation Submenu		Model Validation Submenu ^[81]
Baselines	Store a model branch as a snapshot or baseline Available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions	Ctrl + Alt + B	Baseline ^[81]
Audit View	Display the Audit View, which shows the information that has been recorded by auditing		Audit View ^[302]
Model Import/Export	Display the Import/Export Submenu		Import/Export Submenu ^[81]
Version Control	Display the Version Control Submenu		Version Control Submenu ^[82]
Security	Display the Security Submenu		Security Submenu ^[82]
Use Case Metrics	Set Use Case Metrics to assist in estimating project size		Use Case Metrics ^[404]
Project Statistics	View some basic project statistics		

2.5.3.4.1 Documentation Submenu

Access: **Project | Documentation**

Action	Usage	Shortcut	See also
Rich Text Format (RTF) Report	Generate a report for the currently selected package in Rich Text Format ^[1726]	F8	RTF Documents ^[1726]
HTML Report	Generate a report for the currently selected package in HTML format	Shift + F8	HTML Reports ^[1802]

Action	Usage	Shortcut	See also
Diagrams Only Report	Generate an RTF report containing only diagrams	Ctrl + Shift + F8	Diagrams Only Report ^[1784]
Testing Report	Generate an RTF report of the model's existing tests		Test Documentation ^[1708]
Issues	Generate an RTF report of the model's issues		Project Issues ^[358]
Glossary	Generate an RTF report of the model's Glossary		Generate a (Glossary) Report ^[364]
Implementation Details	Show a list of the elements in the package (selected in the Project Browser) that require implementers, together with any source elements in Realize (Implements) relationships with those elements		Implementation Report ^[1785]
Dependency Details	Show a list of any elements in the package (selected from the Project Browser) that are dependent on another element for their specification		Dependency Report ^[1783]
Testing Details	Show the testing details for the elements in a package, for tests that meet specific criteria		Testing Details Report ^[1788]

2.5.3.4.2 Model Validation Submenu

Access: **Project | Model Validation**

Action	Usage	Shortcut	See also
Validate Selected	Validate a selected element, diagram or package from the Project Browser	Ctrl + Alt + V	Model Validation ^[1688]
Cancel Validation	Cancel the validation process		
Configure	Configure the Validation rules from the list of available rules		Configure Model Validation ^[1689]

2.5.3.4.3 Import/Export Submenu

To perform import and export to XML and CSV.

Access: **Project | Model Import/Export**

Action	Usage	Shortcut	See also
Import Package from XMI	Import a package from an XMI (XML based) file	Ctrl + Alt + I	Import From XMI ^[323]
Export Package to XMI	Export the currently selected package to an XMI (XML based) file	Ctrl + Alt + E	Export To XMI ^[320]
Batch XMI Export	Export a group of controlled packages in one		Batch XMI Export

Action	Usage	Shortcut	See also
	action		[332]
Batch XMI Import	Run a batch import of multiple packages		Batch XMI Import [333]
CSV Import/Export	Import or Export information on Enterprise Architect elements in CSV format	Ctrl + Alt + C	CSV Import ^[342] CSV Export ^[340]
CSV Import/Export Specifications	Set up CSV import Export Specifications		CSV Specifications [338]
Export Reference Data	Export reference data to XML files for convenient model updating		Export Reference Data ^[237]
Import Reference Data	Import reference data from XML files for convenient model updating		Import Reference Data ^[239]

2.5.3.4.4 Version Control Submenu

Access: **Project | Version Control**

Action	Usage	Shortcut	See also
Configure Current Package	Specify whether this package (and its children) is controlled and, if so, which file it is controlled through	Ctrl + Alt + P	Configure Controlled Package [282]
Version Control Settings	Specify the options required to connect to a Source Code Control (SCC) provider		Version Control Settings Dialog ^[270]
Validate Package Configurations	Validate the version control configuration of each package in the model		Validate Package Configurations ^[296]
Resynch Statuses of All Packages	Resynchronize the status of all version controlled packages within your project with the status of the version control provider		Resynchronize the Status of Version Controlled Packages ^[297]
Work Offline	Toggle between 'offline' version control and online version control		Offline Version Control ^[250]

2.5.3.4.5 Security Submenu

To configure security settings for your project.

Access: **Project | Security**

Action	Usage	Shortcut	See also
Manage Users	Add, modify and remove users, including maintaining permissions.		Maintain Users ^[198]
Manage Groups	Add, modify and remove security groups,		Maintain Groups

Action	Usage	Shortcut	See also
	including maintaining permissions.		[203]
Manage Locks	View and manage element locks.		View and Manage Locks [207]
Require User Lock to Edit	Toggle the security policy in force.		Security Policy [197]
Manage My Locks	View and delete your own locks.	Ctrl + Shift + L	Manage Your Own Locks [215]
Login as Another User	Switch the login to a different user ID.		
Change Password	Change current security password.		Change Password [209]
Encrypt Password	Add encryption to your password (prior to Enterprise Architect Release 7.1).		Password Encryption [208]
Enable Security	Enable or disable user security to limit access to update functions in the model.		Enable Security [196]

Notes:

- This feature is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

2.5.3.5 Diagram

The **Diagram** menu enables you to configure diagram properties and options and to save diagram images to file.

Action	Usage	Shortcut	See also
Properties	View and edit the <type> Diagram: <name> dialog for the current diagram	F5	
Find in Project Browser	Locate the current diagram in the Project Browser window	Shift + Alt + G	
Layout Diagram	Automatically layout the current diagram (not available for Behavioral diagrams)		Layout a Diagram Automatically [619]
Switch to	Change the presentation of the diagram to the Diagram List format or the GANTT Chart format		Diagram List [462]
Advanced	Display the diagram Advanced menu		Diagram Advanced Menu [84]
Save	Save the current position of all diagram elements	Ctrl + S	
Save as Image	Save the diagram as a bitmap file (.BMP), Graphics Interchange Format file (.GIF) or Windows Metafile (.WMF)	Ctrl + T	
Copy Image to Clipboard	Copy an image of the current diagram to the clipboard. The image can be in metafile or bitmap format; you set the format on the Options	Ctrl + B	General [422] (page of Options dialog)

Action	Usage	Shortcut	See also
	dialog		
Swimlanes and Matrix	Add, modify and delete swimlanes or the swimlanes matrix for the current diagram		Swimlanes ^[590] Swimlanes Matrix ^[592]
Repeat Last Element	Create an instance of the same type as the last element created	Shift + F3	
Repeat Last Connector	Create an instance of the same type as the last connector created	F3	
Property Note	Display the properties note for the current diagram on screen		Insert Diagram Properties Note ^[587]
Show Grid	Show or hide the diagram grid		
Snap To Grid	Position elements on the diagram grid; there are two options: <ul style="list-style-type: none"> • Standard Grid - constrains elements to the grid when they are added to diagrams • Smart Placement - places elements even distances away from other elements and spaces elements evenly <p>If neither of these options are enabled, the elements can be placed freely on the diagram</p>		
Zoom	Change the zoom factor on the current diagram		Pan and Zoom a Diagram ^[599]

2.5.3.5.1 Diagram Advanced Menu

The Diagram **Advanced** menu enables you to perform a number of very useful operations, such as hiding connectors on the diagram or creating a UML Pattern from the diagram.

Action	Usage	Shortcut	See also
Lock Diagram	Prevent the diagram from being edited, or release the locked diagram for editing This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, if security is enabled		Lock Model Elements ^[210]
Make Model Default	Make the current diagram the default diagram opened when the current model is re-opened (unless you set a User Default diagram, which overrides the model default; see below) This option is also overridden by a project shortcut, which defines displays to present on opening the project To cancel a Model Default diagram, either: <ul style="list-style-type: none"> • Create a dummy diagram, set it as the Model Default and delete it, or • Delete the original diagram (if it is no longer relevant) 		Set the Default Diagram ^[585] Project Shortcuts ^[140]

Action	Usage	Shortcut	See also
Make User Default	<p>(Use in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled)</p> <p>Make the current diagram the default diagram opened when you re-open this model</p> <p>The User Default diagram overrides the Model Default diagram (see Make Model Default, above); this option is itself overridden by a project shortcut, which defines displays to present on opening the project</p> <p>To cancel a User Default diagram, either:</p> <ul style="list-style-type: none"> • Create a dummy diagram, set it as the User Default and delete it, or • Delete the original diagram (if it is no longer relevant) <p>This still blocks the Model Default diagram, whilst Security is enabled; to re-establish the Model Default diagram, set it as the User Default</p>		Project Shortcuts ^[140]
Visible Relations	Hide or show individual connectors for the current diagram	Ctrl + Shift + I	Relationship Visibility ^[749]
Sequence Messages	Change ^[998] the order of the communication messages in the current diagram		Message (Communication Diagrams) ^[996]
Change Type	Change the type of the current diagram		Change Diagram Type ^[581]
Save UML Pattern	Save the current diagram as a UML pattern		Create a Pattern ^[1022]

2.5.3.6 Element

The **Element** menu enables you to configure and access element features and details and control element layout, and to generate documentation and work on source code for the element

Action	Usage	Shortcut	See also
Properties	View the Properties dialog of the selected element	Alt + Enter	Properties Dialog ^[660]
Linked Document	Link the element to a rich text document	Ctrl + Alt + D	Link the element ^[730]
Advanced	Display the element Advanced submenu		Advanced Submenu ^[86]
Find in Project Browser	<p>Find the currently selected element in the Project Browser</p> <p>If no element is selected, find the current diagram in the Project Browser</p>	Alt + G	
Find in Diagrams	Display all occurrences of the currently selected	Ctrl + U	

Action	Usage	Shortcut	See also
	element in any diagram		
Attributes	View and edit the attributes for the selected element	F9	Element attributes ^[696]
Operations	View and edit the operations (methods) for the selected element	F10	Operations ^[708]
Feature Visibility	Select which features and characteristics of the selected element are visible on a diagram	Ctrl + Shift + Y	Visible ^[585]
Inline Features	Display the element Inline Features submenu		Inline Features Submenu ^[87]
Source Code Engineering	Display the element Source Code Engineering submenu		Source Code Engineering Submenu ^[88]
Open Source in External Editor	Open the source code of the selected Class in the default external editor for that language Source code must have been generated, and the selected element must be a Class	F12	The Source Code Viewer ^[1408]
Appearance	Display the element Appearance submenu		Appearance Submenu ^[88]
Alignment	Display the element Position submenus		Position Submenus ^[88]
Make Same			
Z Order			
Size			
Move			
Space Evenly			

2.5.3.6.1 Advanced Submenu

The **Advanced** sub-menu provides various options to choose from to customize the appearance of model elements. Select the **Element | Advanced** menu option to display this submenu.

Action	Usage	Shortcut	See also
Add Tagged Value	Add a Tagged Value to the currently selected element	Ctrl + Shift + T	Quick Add - Tagged Value To Elements ^[763]
Custom References	Show model element cross references	Ctrl + J	Set Up Cross References ^[630]
Rich Text Format (RTF) Report	Generate a report for the currently selected package in rich text format		RTF Documents ^[1728]
Set Parents and Interfaces	Manually set the element's parents or an interface that it realizes	Ctrl + I	Set Element Parent ^[628]
Embedded	Attach elements to the currently selected element	Ctrl + Shift +	Embedded

Action	Usage	Shortcut	See also
Elements		B	Elements Window [654]
Overrides & Implementations	Automatically override methods from parent Classes and from realized interfaces	Ctrl + Shift + O	Override Parent Operations [712]
Change Type	Change the element type of the selected element		Change Element Type [635]

2.5.3.6.2 Inline Features Submenu

Access: **Element | Inline Features**

Action	Usage	Shortcut	See also
Edit Selected	Type over or delete the text of the highlighted feature.	F2	
View Properties	Open the dialog containing details of the selected element feature, or the element if no feature is selected.		
Insert New After Selected	Insert a new item in the element.		
Create Linked Note	Add a Note element linked to the selected item, reflecting the content of that item.		
Add Attribute	Add an attribute to the current element.	Ctrl + Shift + F9	Add an attribute [695]
Add Operation	Add an operation to the element.	Ctrl + Shift + F10	Add an operation [707]
Add Other	Insert other features such as Maintenance and Testing items.	Ctrl + F11	
Delete Selected from Model	Delete the selected item from the model.	Ctrl + Shift+ Delete	

Active keys in this mode include:

Key	Action
Enter	Save current changes.
Ctrl+Enter	Save current changes and open a new slot to add a new item.
Esc	Cancel editing.
Shift+F10	Display the context menu for in-place editing.

2.5.3.6.3 Source Code Engineering Submenu

To forward and reverse engineer code using the language of your choice, select the **Element | Source Code Engineering** menu option.

Action	Usage	Shortcut	See also
Generate Current Element	Generate source code ^[149] for the currently selected element.	F11	Generate source code^[149]
Synchronize Current Element	Synchronize the selected Class with the source code.	F7	
Batch Generate Selected Element (s)	Batch generate source code for the currently selected element(s).	Shift + F11	
Batch Synchronize Selected Element (s)	Batch synchronize the currently selected element (s) with source code.	Ctrl + R	
Open Source Directory	Open the directory containing the source for this element.	Ctrl + Alt + Y	

2.5.3.6.4 Appearance Submenu

The **Appearance** sub-menu provides various options to choose from to customize the appearance of model elements.

Action	Usage	Shortcut	See also
Default Appearance	Set border, font and background color and border thickness for the selected element, as its default appearance ^[404] .	Ctrl + Shift + E	Default appearance^[64]
Autosize	Auto-size a group of elements in a diagram to a best fit.	Alt + Z	
Set Font	Change the font ^[136] of the text displayed on the element in a diagram.		Text font^[65]
Alternate Image	Select an alternative image for the selected element.	Ctrl + Shift + W	
Apply Image From Clipboard	Insert the image currently held on the clipboard.		

2.5.3.6.5 Position Submenus

These Element menu submenus enable you to size and position elements on the diagram, relative to each other.

The Alignment Sub-Menu:

Use the **Alignment** sub-menu to align the selected element(s) to each other.

Action	Usage	Shortcut	See also
Left	Align left edges of elements.	Ctrl + Alt + Left	
Right	Align right edges of elements.	Ctrl + Alt + Right	
Top	Align top edges of elements.	Ctrl + Alt + Up	
Bottom	Align bottom edges of elements.	Ctrl + Alt + Down	
Centers	Align centers of elements, horizontally or vertically.		

The Make Same Sub-Menu:

Use the **Make Same** sub-menu to make the selected elements the same width, the same height or both.

Action	Usage	Shortcut	See also
Width	Make all selected elements the same width		
Height	Make all selected elements the same height		
Both	Make all selected elements the same height and width		

The Z Order Sub-Menu:

Use the **Z Order** sub-menu to move the selected element(s) back, forward, to the front of all other elements or behind all other elements.

Action	Usage	Shortcut	See also
Forward	Move selected element forward one position in the Z order		Z Order Elements [584]
Back	Move selected element back one position in the Z order		
To Bottom	Send selected element behind all other elements		
To Top	Bring selected element in front of all other elements		

The Size Sub-Menu

Use the **Size** sub-menu to make the selected element(s) wider, narrower, taller or shorter by one increment.

Action	Usage	Shortcut	See also
Wider	Make all selected elements wider	Ctrl + →	
Narrower	Make all selected elements narrower	Ctrl + ←	
Shorter	Make all selected elements shorter	Ctrl + ↑	

Action	Usage	Shortcut	See also
Taller	Make all selected elements taller	Ctrl + ↓	

The Move Sub-Menu

Use the **Move** sub-menu to move the selected element(s) left, right, up or down by one increment.

Action	Usage	Shortcut	See also
Right	Move all selected elements to the right	Shift + →	
Left	Move all selected elements to the left	Shift + ←	
Up	Move all selected elements upwards	Shift + ↑	
Down	Move all selected elements downwards	Shift + ↓	

The Space Evenly Sub-Menu

Use the **Space Evenly** sub-menu to distribute the selected elements evenly.

Action	Usage	Shortcut	See also
Across	Space elements evenly, horizontally.	Alt + -	
Down	Space elements evenly, vertically.	Alt + =	

2.5.3.7 Tools

The **Tools** menu provides access to various tools including those related to code engineering, model management, spell checking, customization of features and setting operational options.

Action	Usage	Shortcut	See also
Open Source File	Open any type of external source file (code, XML, DDL) for editing	Ctrl + Alt + O	The Source Code Viewer ^[1409]
Source Code Engineering	Display the Source Code Engineering submenu		Source Code Engineering ^[91]
Database Engineering	Display the Database Engineering submenu		Database Engineering ^[92]
Web Services	Display the Web Services submenu		Web Services ^[92]
XML Schema	Display the XML Schema submenu		XML Schema ^[92]
Model Transformation (MDA)	Display the Model Transformation submenu		Model Transformation ^[93]
Spelling Tools	Configure and run the spelling checker		Configure and Run ^[93]
Data Management	Manage your project's data		Data Management

Action	Usage	Shortcut	See also
			Submenu ^[93]
MDG Technology Import	Import an MDG Technology file to APPDATA (Not available in the Desktop edition)		Import ^[1035] a file
Generate MDG Technology File	Display the MDG Technology Wizard (Not available in the Desktop edition)		MDG Technology Wizard ^[1064]
Wordpad	Open Wordpad		
Windows Explorer	Open Windows Explorer (different options might be listed, to open other applications)		
Customize	Customize the operation of Enterprise Architect		The Customize Dialog ^[118]
Options	Customize your general settings through the Options dialog	Ctrl + F9	Options dialog ^[422]

2.5.3.7.1 Source Code Engineering Submenu

To forward and reverse engineer code using the language of your choice select the **Tools | Source Code Engineering** menu option (Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions).

Action	Usage	Shortcut	See also
Generate Package Source Code	Generate source code for the currently selected package	Ctrl + Alt + K	Generate source code ^[1493]
Synchronize Package Contents	Synchronize the selected package with the source code	Ctrl + Alt + M	
Import Source Directory	Import and reverse engineer an entire directory structure	Ctrl + Shift + U	Import ^[1513] an directory structure
Import Binary Module	Import a binary module		Import a binary module ^[1513]
Import ActionScript Files	Import code written in ActionScript with the file extension .AS		ActionScript ^[1511] code
Import C Files	Import code written in ANSI C with the file extension .C or .H		ANSI C ^[1511] source code
Import C# Files	Import code written in the C# programming language with the file extension .CS		C# ^[1511] source code
Import C++ Files	Import code written in the C++ programming language with the file extension .H, .HPP or .HH		C++ ^[1511] source code
Import Delphi Files	Import code written in the Delphi programming language with the file extension .PAS		Delphi ^[1511] source code
Import Java Files	Import code written in the Java programming language with the file extension .JAVA		Java ^[1511] source code

Action	Usage	Shortcut	See also
Import PHP Files	Import code written in PHP with the file extensions .INC, .PHP, .PHP4		PHP ^[1512] source code
Import Python Files	Import code written in Python with the file extension .PY		Python ^[1512] source code
Import Visual Basic Files	Import code written in the Visual Basic programming language with the file extension .FRM, .CLS, .BAS or .CTL		Visual Basic ^[1512] source code
Import VB.NET Files	Import code written in the VB.NET programming language with the file extension .VB		VB.NET ^[1512] source code

2.5.3.7.2 Database Engineering Submenu

Access: **Tools | Database Engineering**

Action	Usage	Shortcut	See also
Import DB Schema from ODBC	Import a database schema from an ODBC data source		Import a database schema from an ODBC ^[1375]
Generate Package DDL	Generate a Package DDL script to create the tables in the currently selected package		Generate a Package DDL ^[1380]

2.5.3.7.3 Web Services Submenu

Access: **Tools | Web Services**

Action	Usage	Shortcut	See also
Import WSDL	Reverse engineer ^[1622] a Web Service Definition Language (WSDL) file as a UML Class model		Reverse engineer ^[1622]
Generate WSDL	Forward ^[1621] engineer a UML Class model to a WSDL file		Forward ^[1621] engineer

2.5.3.7.4 XML Schema Submenu

Access: **Tools | XML Schema**

Action	Usage	Shortcut	See also
Import XML Schema	Reverse engineer a W3C XML Schema (XSD) file as a UML Class model		Reverse ^[1600] engineer
Generate XML Schema	Forward engineer a UML Class model to a W3C XML Schema (XSD) file		Forward ^[1599] engineer

2.5.3.7.5 Model Transformation (MDA) Submenu

Access: **Tools | Transformations**

Command	Action	Shortcut	See also
Transform Selected Elements	Perform an MDA-style transformation on the currently selected elements	Ctrl + H	Selected elements [1305]
Transform Current Package	Perform an MDA-style transformation on the currently selected package	Ctrl + Shift + H	Selected package [1305]

2.5.3.7.6 Spelling Tools

Access: **Tools | Spelling Tools**

Action	Usage	Shortcut	See also
Spell Check Project	Spell check the whole project	Ctrl + F7	Spell check [376]
Spell Check Current Package	Spell check the current package, selected in the Project Browse	Ctrl + Shift + F7	Spell check [376]
Spelling Options	Specify a range of options to configure how the spell checker operates		Configure options [373]

2.5.3.7.7 Data Management Submenu

Use the options on the **Data Management** submenu to manage your project's data.

Action	Usage	Shortcut	See also
Project Transfer	Move a complete project from one repository to another You cannot move a project from a source .EAP file of a version earlier than 3.5.0		Perform a Project Data Transfer [343]
Reset IDs	Reset the table auto increment column values		Reset Table Auto Increment or Identity Columns [416]
Project Compare	Compare the total project sizes of two projects		Compare Projects [337]
Project Integrity Check	Check the data integrity of a project	Shift + F9	Check Project Data integrity [414]
Run Patch	Execute an SQL Patch		Execute an SQL Patch [418]
Manage .EAP File	Repair, compact or replicate your .EAP file		Repair, compact or replicate [94]

2.5.3.7.7.1 Manage .EAP File Submenu

Use the options on this submenu to repair, compact or replicate your .EAP file.

Action	Usage	See also
Repair .EAP File	Repair an Enterprise Architect project If a project has not been closed properly, in rare cases it might not open correctly; this option attempts to repair such projects All users must be logged off the project while it is being repaired	Repair ^[419] a project
Compact .EAP File	Compact an Enterprise Architect project Eventually projects might benefit from compacting to conserve space Ensure everyone is logged off the target project, then select this option to compact the project.	Compact ^[418] a project
Make Design Master	Make a design master project; this is the master project for creating replicas	Make a design master ^[190] project
Create New Replica	Create a new replica from the Design Master	New replica ^[188] Design Master ^[190]
Synchronize Replicas	Copy changes from one replica set member to another	Copy changes ^[197]
Remove Replication	Remove all replication features if you no longer require a model to be replicable	Remove ^[192] all replication features
Resolve Replication Conflicts	Resolve any conflicts caused when multiple users have changed the same element between synchronization points	Resolve any conflicts ^[193]

2.5.3.8 Analyzer

The **Analyzer** menu enables you to perform build, debug and profiling operations on your model code.

Action	Usage	Shortcut	See also
Execution Analyzer	Display the Package Scripts window, which enables you to create and export Package Build Scripts.		
Build	Build ^[1415] the application for your current script, and execute the Build command in the Build Scripts dialog.	Ctrl + Shift + F12	Build ^[1415] the application
Debug	Display the Debug ^[95] menu, which provides a set of debugging commands.		Debug ^[95] menu
Record	Display the Record ^[96] menu, which provides various options for recording a debug session.		Record ^[96] menu
Profile	Display the Profiler ^[1667] window, which enables you to report on runtime aspects such as what functions are most-frequently called in an application, which tasks are taking longer than		Profiler ^[1658] window

Action	Usage	Shortcut	See also
	expected, and which tasks are taking the longest time to execute.		
Test	Display the Test submenu, which provides access to the Testpoint Manager and executing a Test script		Test submenu
Simulation	Display the Simulation window, which enables you to simulate the debugging of an application during the design of that application		
Breakpoints & Markers	Display the Breakpoints & Markers window, which you use to manage breakpoints and markers for debugging.		Manage breakpoints and markers
Call Stack	View the currently-running threads in a debugging session, on the Call Stack.		Capture Current Work Environment
Locals	View the local variables and their values for the current local thread, on the Locals window		Locals window
Watches	Evaluate data items that are not available as local variables - such as data items with static Class member items, on the Watches window		Watches window
More Analyzer Windows	Display the More Analyzer Windows submenu, which provides access to the Package Build Scripts dialog, the Workbench window, the Modules window and the Memory Viewer window		More Analyzer Windows submenu
Run	Execute the Run command you configured in the Build Scripts dialog.	Ctrl + Alt + N	Run command
Deploy	Execute the Deploy command you configured in the Build Scripts dialog.	Ctrl + Shift + Alt + F12	Deploy command

2.5.3.8.1 Debug Menu

The **Debug** menu provides a range of options for running a debug session.

Action	Usage	Shortcut	See also
Debugger	Display the Debugger window, which enables you to control debugging and shows output from a debug session	Alt + 8	
Debug Run	Start or resume the debugger If no debug session is in progress, the active Analyzer Script is used to determine which application to run or whether the debugger should attach to an already running process	F6	
Debug Pause	Suspend execution of process being debugged		
Step Into	Step into the current function	Shift + F6	Step Into Function Calls
Step Over	Step over the current function	Alt + F6	Step Over Lines of

Action	Usage	Shortcut	See also
			Code ^[1448]
Step Out	Step out of the current function	Ctrl + F6	Step Out of Functions ^[1449]
Debug Stop	Stop the current debug session	Ctrl + Alt + F6	Start & Stop Debugger ^[1447]
Generate Sequence Diagram From Stack	Create a Sequence diagram from the Stack Trace History		Create Sequence Diagram Of Call Stack ^[1450]

2.5.3.8.2 Record Menu

The **Record** menu provides various options for recording a debug session.

Action	Usage	Shortcut	See also
Recorder	Display the Record & Analyze window, which enables you to review the information recorded during a debug session		The Recording History ^[1651]
Start Debug Recording	Enabled when the program is at a breakpoint Enter manual record mode; each time you step into a function, the call is captured When you step out, the return is captured		
Stop Debug Recording	Enabled when recording is in progress Stops recording		
Auto Record Thread	Enabled when the program is at a breakpoint The thread that is halted is recorded, and the program resumes execution; whenever the thread makes a function call or a function call exits, this information is captured to the History window The Stack Trace History, Stack tab and Source Code Editor dynamically update to reflect the current execution sequence for the thread; Stack Trace Recording ends when the thread ends or when you click on the Stop button		
Show/Hide Execution	Display the executing code while the recording is in progress; this slows recording down considerably, and is not used when recording is being controlled by recording markers The Call Stack is updated in realtime, even while recording is in progress Watching this window while recording can give a quick picture of what is going on, even without producing a Sequence diagram		View the Call Stack ^[1447]
Generate Sequence	Create a Sequence diagram from the Record & Analyse window		Generate a Diagram ^[1652]

Action	Usage	Shortcut	See also
Diagram from Recording History			

2.5.3.8.3 Test Menu

This submenu provides access to the Testpoint Manager and enables you to run test scripts.

Action	Usage	Shortcut	See also
Testpoint Manager	Display the Testpoints window to execute test scripts and monitor or control the output		The Testpoints Window ^[1676]
Run Test Script	Execute the Test command you configured in the Build Scripts dialog	Ctrl + Alt + T	Add Testing Command ^[1451]

2.5.3.8.4 More Analyzer Windows

The **More Analyzer Windows** menu provides access to additional analysis tools.

Command	Action	Shortcut	See also
Analyzer Scripts	Create and configure Package Build compiler scripts.	Shift + F12	Managing Analyzer scripts ^[1392]
Workbench	Display the Workbench window, which enables you to create your own workbench variables and invoke methods on them.		Object Workbench ^[1663] Create workbench variables ^[1665] Invoke methods ^[1666]
Modules	Review details of the loaded modules, on the Modules window.		Show Loaded Modules ^[1446]
Memory Viewer	Review a process memory debugging session, using the Memory Viewer window.		Inspect Process Memory ^[1445]

2.5.3.9 Addins

The **Add-Ins** menu enables you to connect to, display information on, work with and manage your Add-Ins.

Action	Usage	Shortcut	See also
<Add-In Name(s)> >	Access the facilities of the selected Add-In, through an Add-In submenu (below). For example, if you have TOGAF enabled on your system, you might click on the TOGAF menu option and display a submenu of options available to the TOGAF user in Enterprise Architect.		
Add-In Windows	Display a window, or list of windows, provided by any Add-Ins you have installed and enabled. If no windows are provided, displays an empty, docked Add-Ins window.		

Action	Usage	Shortcut	See also
Manage Add-Ins	Display the Manage Add-Ins [412] dialog, which you use to enable or disable Add-Ins for use. Any technology loaded by an Add-In is automatically enabled; if you do not want to use it, you can disable it on the dialog.		Manage Add-Ins [1974] dialog

Add-In Submenu

Action	Usage	Shortcut	See also
<Add-In specific options>	List options to perform various functions specific to the Add-In. For example, the MDG Technology For Zachman Framework, as an Add-In, has the options Open Example Model and Insert New Framework Model.		
Help	Display the Help subsystem for the Add-In.		
About	Display information on the Add-In installation, such as version, registration details and copyright statement.		

2.5.3.10 Settings

The **Settings** menu enables you to configure various settings for your overall project, such as stereotypes, Tagged Values, cardinality values, datatypes, language macros, local directories, image management.

Action	Usage	Shortcut	See also
UML Types	Configure stereotypes [773] , Tagged Value Types [775] and the cardinality list [776] for your project.		Stereotypes [773], Tagged Value Types [775], cardinality list [776]
Project Types	Display the Project Types [803] submenu.		Project Types [99]
Namespace Roots	Locate and delete model namespaces [1495] .		Model namespaces [1495]
Project Template Package	Configure or change the default element template directory.		
Auto Names and Counters	Configure automatic naming [628] for elements.		Configure automatic naming [628]
MDG Technologies	Display the MDG Technologies [1031] dialog, which enables you to activate and use MDG Technology files.		MDG Technologies [1031]
Code Generation Templates	Modify code generation templates using the Code Templates Editor [1482] .	Ctrl + Shift + P	Code Templates Editor [1482]
Local Directories and Paths	Configure local directories and paths [1523] .		Local directories and paths [1523]

Action	Usage	Shortcut	See also
Code Datatypes	Add, modify and delete programming languages datatypes [777] .		Programming languages datatypes [777]
Preprocessor Macros	Add and delete preprocessor macros [1391] .		Preprocessor macros [1525]
Database Datatypes	Add, modify and delete database datatypes [1278] .		Database datatypes [1353]
Transformation Templates	Modify transformation templates using the Transformation Templates Editor [1330] .	Ctrl + Alt + H	Transformation Templates Editor [1330]
Images	Open the Image Manager [1785] .		Image Manager [593]
Colors	Configure the custom colors for the project. There are two options: <ul style="list-style-type: none"> • Get Project Custom Colors - get the custom colors • Set Project Custom Colors - set the custom colors. Custom colors are as used in the Appearance [576] dialog.		Appearance [641] dialog

2.5.3.10.1 Project Types Submenu

The **Project Types** submenu enables you to create and manage project variables and definitions for use across the project.

Action	Usage	Shortcut	See also
People	Display the People dialog, which enables you to configure the authors, clients, resources and roles for your project.		People [778] dialog
General Types	Display the General Types dialog, which enables you to configure requirements, status types, constraints and scenarios for your project		General Types [782] dialog
Project Indicators	Define the project indicators (risks, efforts and metrics) used in Resource Management		Risks [355] Efforts [353] Metrics [354]
Maintenance	Display the Maintenance dialog, which enables you to track problem types and test types		Maintenance [787]
Estimation Factors	Display the Estimation factors dialog, which enables you to configure estimation factor types (Technical Complexity Factors, Environmental Complexity Factors, and Default Hour Rate) for your project		Estimation [400] factor Technical Complexity Factors [401] Environmental Complexity Factors [402]

Action	Usage	Shortcut	See also
			Default Hour Rate [404]

2.5.3.11 Window

The **Window** menu provides access to various actions related to configuring open windows.

Action	Usage	Shortcut	See also
Full Screen	Reset the display to full screen so that only the work area and main menu are shown - no toolbars or windows To return to your normal working display, either click on the Full Screen option again or click on the Close Full Screen pop-up menu option You can also restore individual menus and toolbars using the View menu options		
Save All Modified	Save all modified data		
Reload Current View	Refresh the current view		Refresh the current view [187]
Set Focus to Current View	Make the current view the active one, so that key strokes immediately act on that view	Ctrl + Shift + 0	
Close Current View	Close the current view		
Close All Except Current	Close all except the currently selected view		
Close All	Close all opened windows in the main tab view		
Autohide Active Window	Autohide [105] the window that currently has focus	Ctrl + Shift + F4	Autohide [105] windows
Autohide All Docked Windows	Autohide [105] all windows that are docked		Autohide [105] windows
Close Active Window	Close the window that currently has focus	Ctrl + F4	
Show Start Page	Show or hide the Start Page The setting of this option persists through shut down and restart, until you specifically change the option again		Start Page [70]
Always on Top	Force the main Enterprise Architect window to be on top of all other windows		

2.5.3.12 Help

The **Help** menu provides access to the Enterprise Architect help files, the Read Me file, the Enterprise Architect End User License Agreement and various features on the [Sparx Systems website](#).

Action	Usage	Shortcut	See also
About EA	View information about Enterprise Architect, including your registration details		
Help Contents	Display the Enterprise Architect Help, starting at the introductory page		
On-Line Resources	Display the Online Resources submenu; see below		
Open Example Model	Open the <i>EA.Example</i> model provided with the Enterprise Architect installer		
Register and Manage License Key(s)	Configure and manage the license keys used to register the name and keys for Enterprise Architect and its Add-Ins		License Management ^[2094]
View License Agreement	View the Enterprise Architect End User License Agreement		
Ordering Information	View information on how to purchase Enterprise Architect		Order Enterprise Architect ^[26]
Read Me	View the <i>Readme.txt</i> file, which details the changes and enhancements in Enterprise Architect, build by build		
Keyboard Accelerator Map	View the keyboard accelerator map You can customize your keyboard shortcuts, if required		Customize your keyboard shortcuts ^[124]
EA on the Web	Visit the Sparx Systems website		Sparx Systems website

The On-Line Resources Sub-Menu

Access help and resources on-line at the Sparx Systems website.

Action	Usage	Shortcut	See also
User Forum and News	Visit the Enterprise Architect user forum.		User forum
Request-a-Feature	Request a feature you would like to see in Enterprise Architect.		Request a feature
Bug Report Page	Report the details of a bug you have found in Enterprise Architect.		Report the details of a bug
Latest Version Details	Display the Announcements folder of the user forum, providing details of the latest Enterprise Architect build.		Details of the latest Enterprise Architect build

Action	Usage	Shortcut	See also
Automation Interface	Access the Enterprise Architect Automation Interface pages on the Sparx Systems website.		Automation Interface
Introducing UML	Access the Sparx Systems online UML tutorials.		UML tutorials
Pricing and Purchase Options	Purchase or upgrade Enterprise Architect over the internet.		Purchase or upgrade Enterprise Architect

2.5.4 Standard Windows

There are many dockable windows available to use in Enterprise Architect. These can be accessed either:

- Through the **View** menu, or
- Through the context menu accessed by right-clicking on the main menu

Having displayed a window, you can also reduce it to a tab off to one side, or autohide it.

The dockable windows available include:

Topic	Link
Project Browser	Project Browser ^[44]
Properties	Properties ^[68]
Project Information	The Project Information Window ^[35]
Testing	Testing ^[169]
Maintenance	Maintenance ^[171]
Toolbox	Toolbox ^[54]
Resources	Resources ^[78]
Source Code Viewer	Source Code Viewer ^[140]
Scripting window	Scripting window ^[181]
Debug	Debug ^[144]
Learning Center	Learning Center ^[72]
Notes	Notes ^[76]
Traceability	Traceability ^[49]
Tagged Values	Tagged Values ^[76]
Project Management	Project Management ^[34]
Model Views	Model Views ^[46]
Element Browser	Element Browser ^[68]
Relationships	Relationships ^[50]
Scenarios & Requirements	Scenarios & Requirements ^[68]
Pan & Zoom	Pan & Zoom ^[47]

Topic	Link
Layout Tools	Layout Tools ^[604]
Team Review	Team Review ^[216]
Diagram Filters.	Diagram Filters ^[489]

Learn More:

- [Docking Windows](#) ^[103]
- [Autohide Windows](#) ^[105]

Notes:

- On the Testing, Maintenance and Project Management windows, any descriptive, history, input or results text for a selected item is also displayed in the Notes window; you cannot edit this text in the Notes window

2.5.4.1 Dock Windows

A number of Enterprise Architect windows can be freely positioned on the screen, or docked against any edge of the *application workspace*. These windows are collectively called dockable windows.

Drag the window around the application workspace until you find a comfortable way of working. The examples below describe a few ways you can rearrange the windows to suit your work habits.

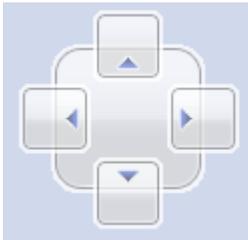
Floating Windows

To float a window anywhere on the screen, just drag the window by its title bar to the required position.

Dock a Window Against an Edge

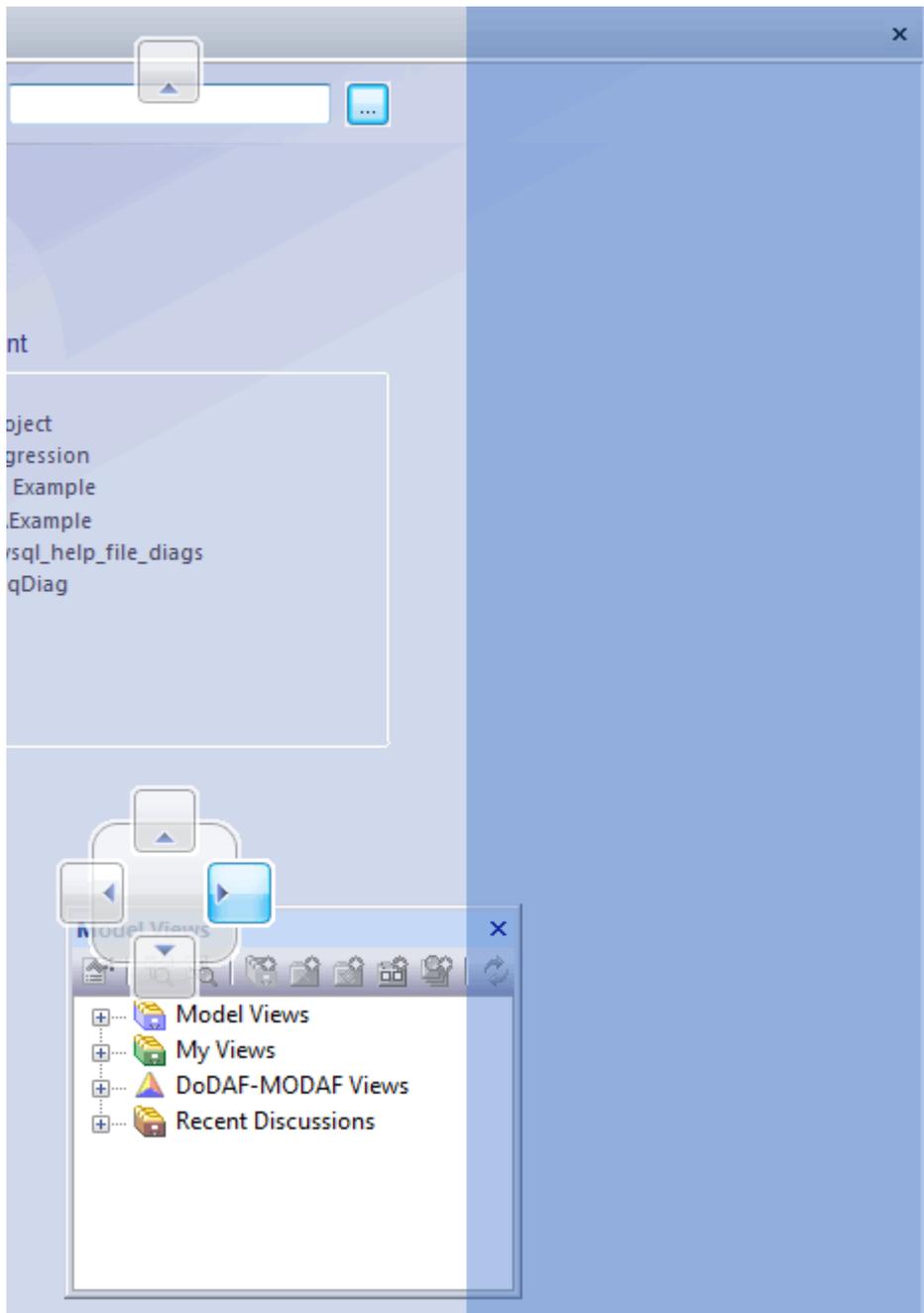
The *navigation compass* enables you to dock windows against an edge of the application workspace. You drag the window over one of the points of the compass to dock it into a tabbed location. The window does not overlap any other window, so if you are docking several windows you could cover the workspace; however, you can avoid this by combining them in a single tabbed frame.

To dock a window against an edge, follow the steps below

Step	Action	See also
1	Click on the item to move and start dragging it towards the required position. This activates the navigation compass. 	
2	Drag the window onto a compass point. The screen display shades the area where the window is to be placed.	

Step	Action	See also
3	Release the mouse button over the compass point to confirm the position; this docks the window.	

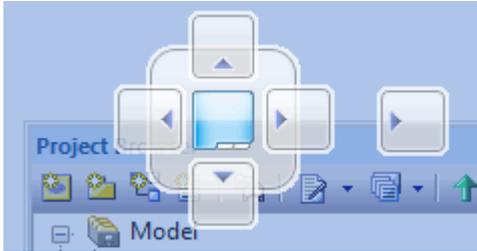
In the example below, when the mouse button is released the **Model Views** window is docked into the shaded area.



Dock Required Windows into One Frame

You can also dock all of the windows you are using into a single frame, by either:

- Dragging the title bar of each window up to the title bar of the first docked window, or
- Dragging each window over the 'tabbed frame' icon in the middle of the compass, when another window is already docked.



You can do this with all dockable windows.

To separate a window from a combined frame, click on the window tab at the bottom of the frame and drag it away.

Learn More:

- [Dockable Windows](#) ⁽¹⁰²⁾

2.5.4.2 Autohide Windows

Autohide Using the Toggle Button

- You can automatically hide browser frames by clicking on the  button, located in the top right corner of the frame.
- To turn off the autohide for a particular set of windows within a frame, click on the  button.

Use Automatically Hidden Windows

When you automatically hide a set of windows in a frame, the tabs contract to the outside of the application workspace.



Hover the cursor over a window symbol to expand both the tab and the associated window.

You can also use the **View | Visual Style | Animate Autohide Windows** menu option to animate windows that have been automatically hidden.

2.5.5 Standard Toolbars

Enterprise Architect provides a set of toolbars containing convenient shortcuts to a wide range of common functions that you can perform on your project as a whole through to the individual modeling units. You can also customize the toolbars by deleting and reordering the default button set.

The toolbars are initially displayed in the toolbar banner at the top of the workspace, but you can display them at the foot of the workspace, drag and dock them within the workspace, or float them over the application; this is useful when you are using a certain set of functions a lot in a particular area.

You can also dock toolbars to the edge of the Enterprise Architect workspace by dragging them by the title bar and placing them against the appropriate edge.

The toolbars available include:

Topic	Link
Default Tools Toolbar	Default Tools Toolbar ^[107]
Project Toolbar	Project Toolbar ^[108]
Code Generation Toolbar	Code Generation Toolbar ^[109]
UML Elements Toolbar	UML Elements Toolbar ^[110]
Diagram Toolbar	Diagram Toolbar ^[111]
Current Element Toolbar	Current Element Toolbar ^[112]
Current Connector Toolbar	Current Connector Toolbar ^[113]
Format Toolbar	Format Toolbar ^[54]
Workspace Layouts Toolbar	Workspace Layouts Toolbar ^[115]
Status Bar	Status Bar ^[117]
Notes Toolbar	Notes Toolbar ^[770]

Each toolbar has a drop-down arrow at the right-hand end, , which can be enabled or hidden from the Visual Style Sub-Menu. If you click on this drop-down arrow, the **Add or Remove Buttons** option displays. Select this option to show a context menu listing the toolbars that are currently displayed, and an option to customize both your own toolbars and the system-provided toolbars.

You can select one of the toolbars identified on the context menu to list the icons available through that toolbar. Click on the icons as necessary to hide or show them in the toolbar.

Learn More:

- [Customize Commands](#)^[118]
- [The Visual Style Sub-menu](#)^[79]
- [The Customize Window](#)^[118]

2.5.5.1 Default Tools Toolbar

The Default Tools toolbar provides instant access to the most commonly used tools in Enterprise Architect including saving, printing and technology selection.

Access: [View](#) | [Toolbars](#) | [Default Tools](#)

Use To:

- Create new projects and open existing projects
- Save changes to the current diagram
- Edit, cut, copy and paste diagram objects
- Print preview, configure print settings and print the current diagram
- Open the Package Browser
- Define, manage and run Model Searches
- Select the working Technology for toolbox and element creation
- Access Enterprise Architect Help

Reference:



Action	Usage	Shortcut	See also
New Project	Create a new Enterprise Architect project (as EAP file)	Ctrl + N	New Enterprise Architect Project ^[145]
Open Project	Open an existing project	Ctrl + O	Open a project ^[138]
Save Diagram	Save the current diagram	Ctrl + S	
Cut	Cut selected element(s) from diagram and copy to Enterprise Architect clipboard	Ctrl + X	
Copy	Copy selected element(s) from diagram to Enterprise Architect clipboard	Ctrl + Space (Ctrl + C)	
Paste	Paste from Enterprise Architect clipboard as reference on current diagram	Shift + Insert (Ctrl + V)	
Undo	Undo last action	Ctrl + Z	Undo last action ^[603]
Redo	Redo last undone action	Ctrl + Y	Redo last action ^[603]
Print Preview	Display the current diagram as it will appear when printed		
Page Setup	Configure print settings for the current diagram		
Print	Print the current diagram	Ctrl + P	
Package Browser	Displays current package as a list of editable elements		Package Browser ^[456]

Open Model Search	Model Search window - define, manage and run model searches	Ctrl + Alt + A	Use model search ^[475]
Technology List	Sets the active technology for the Toolbox and the diagram space bar context element menu		Manage MDG Technologies ^[1035]
Help Contents	Access the Enterprise Architect Help	F1	Help Menu ^[107]

Notes:

- The **Cut** and **Copy** buttons are only enabled when an item is selected on the current diagram
- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Default Tools** menu option

2.5.5.2 Project Toolbar

The Project toolbar provides tools for performing routine tasks such as creating packages, diagrams and elements, searching the model and generating documentation.

Access: **View | Toolbars | Project**

Use to:

- Reload current project
- Create new diagrams, packages and elements
- Perform model searches
- Generate documentation
- Examine project issues
- Review glossary terms
- Set project options

Reference:

Action	Usage	Shortcut	See also
Reload Project	Reload the current project	Ctrl + Shift + F11	Reload current project ^[187]
New Diagram	Create a new diagram in currently selected package	Ctrl + Insert	
New Package	Create a new package within the currently selected package	Ctrl + W	
New Element	Create a new model element under the currently selected package or element	Ctrl + M	
Search Browser	Run a simple search within the project browser	Ctrl + Shift + F	
Search Model	Search the entire project using pre-defined and customized searches	Ctrl + F	Model Search ^[480]

RTF Reports	Define and generate RTF project reports	F8	
Project Issues	Review project issues		
Project Glossary	Define glossary terms		
Set Options	Configure display and behaviour options		

Notes:

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Project** menu option

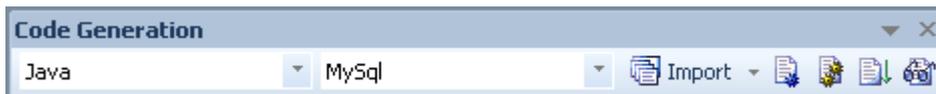
2.5.5.3 Code Generation Toolbar

The Code Generation toolbar can import, generate, synchronize and open source code. Convenient menus are provided to set the default language and default database type for each Project.

Access: **View | Toolbars | Code Generation**

Use to:

- Set the default language
- Set the default database type
- Import Classes and Interfaces from source files
- Generate code for a single selected Class
- Batch generate code for one or more selected Classes
- Synchronize selected Classes with source code
- View code in default editor

Reference:

Action	Usage	Shortcut	See also
Set Default Language	Set the language for all newly created elements		
Set Default Database Type	Set the database type for all newly created database elements		
Import	Choose files to import for reverse engineering source code into the current diagram. Use the drop-down menu to override the project's default language for a particular import.		

			
Generate Source Code	Generate Source code for the currently selected element	F11	
Batch Generate Source Code	Generate Source code for multiple selected elements	Shift + F11	
Synchronize Element with Source Code	Synchronize selected element with its associated source code	F7	
Open Source Code	Open the source code of the selected Class in the default external editor for that language Source code must have been generated, and the selected element must be a Class	F12	The Source Code Viewer ^[1408]

Notes:

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Code Generation** menu option

2.5.5.4 UML Elements Toolbar

The UML Elements toolbar can quickly and easily insert UML elements onto a diagram, including Notes, text, hyperlinks and diagram specific UML elements.

Access: **View | Toolbars | UML Elements**

Use to:

- Insert a variety of elements onto the current diagram

Reference:

Align Left	Align selected elements to the left	Ctrl + Alt + ←	
Align Right	Align selected elements to the right	Ctrl + Alt + →	
Align Top	Align selected elements to the top	Ctrl + Alt + Up Arrow	
Align Bottom	Align selected elements to the bottom	Ctrl + Alt + ↓	
Z Order Top	Bring selected element to top of Z order		Diagram Context Menu ⁵³⁸
Z Order Bottom	Move selected element to bottom of Z order		Diagram Context Menu ⁵³⁸
Previous diagram	Go to previous diagram	Alt + ←	
Next Diagram	Go to next diagram	Alt + →	
Default Diagram	Go to default diagram		
Zoom In	Zoom In		
Zoom Out	Zoom Out		
Zoom to fit diagram	Zoom to fit diagram		
Zoom to fit page	Zoom to fit page		
Zoom to 100%	Zoom to 100%		
Diagram Auto-layout	Auto-layout diagram (not for Behavioral diagrams) in the Digraph layout		Digraph ⁶¹² layout
Diagram Properties	Show diagram properties	F5	
Paste Appearance	Paste appearance as copied into the Painter from an element's Appearance context menu		Appearance ⁵⁷⁶ context menu
Delete	Delete selected element(s)	Ctrl + D	

Notes:

- Any actions that result in a change in diagram content and appearance (including Zoom) are saved as changes to the diagram
- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Diagram** menu option

2.5.5.6 Current Element Toolbar

The Current Element toolbar performs common operations on a selected element such as viewing properties, operations, attributes. It is also capable of specifying the visibility of element features and compartments and locking an element.

Access: [View](#) | [Toolbars](#) | [Current Element](#)

Use to:

- View and modify element properties
- Set an element's parent or implement interfaces
- View and modify Operations
- View and modify Attributes
- Specify the visibility of element features and compartments
- Specify the run state of an element (or, for Parts, property value)
- View use of element in other structures such as diagrams
- Locate the element in the Project Browser window
- View the cross reference list for this element
- Lock or unlock the current element
- Add a Tagged Value to the current element

Reference:



Action	Usage	Shortcut	See also
Edit Properties	View and edit properties of the selected element	Alt + Enter	
Set Parent	Set the current element's parent or implement interfaces	Ctrl + I	
Operations	Create and edit operations for the selected element	F10	
Attributes	Create and edit attributes for the selected element	F9	
Show Features	Specify which compartments and information to display for the selected element in the current diagram	Ctrl + Shift + Y	
Set Run State	Set the element run state or attribute initializers	Ctrl + Shift + R	
Find Element Usage	Find all diagrams in which the selected element is used.	Ctrl + U	
Find in Project Browser	Find the selected element in the Project Browser	Alt + G	
Configure References	Set and view custom references to other elements and diagrams	Ctrl + J	
Lock Element	Apply or release a lock on the selected element		Lock Model Elements ^[210]
Add Tagged Value	Add a tagged value to the selected element	Ctrl + Shift + T	

Notes:

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Current Element** menu option

2.5.5.7 Current Connector Toolbar

The Current Connector toolbar can easily modify the properties and style of a connector, show and hide labels/connectors and pin each end.

Access: **View | Toolbars | Current Connector**

Use to:

- View and modify properties for the current connector
- Set the connector line style
- Attach a note to the currently selected connector
- Set the visibility for labels of the connector
- Set the visible or hidden relations in the current diagram
- Reverse the direction of the currently selected connector
- Pin the start and/or connector ends to a position on the target element (drop menu)

Reference:

Action	Usage	Shortcut	See also
Edit Properties	View and edit properties of the selected connector	Enter	
Set Style	Set the connector line style - choose from list		
Attach Note	Attach a note or constraint to one or more connectors in the current diagram		
Show Labels	Set label visibility for the selected connector		
Show Relationships	Show and hide specific relationships within the current diagram	Ctrl + Shift + I	
Reverse Direction	Reverse the direction of the selected connector		
Pin Connector Ends	Pin end points of the selected connector		

Notes:

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Current Connector** menu option

2.5.5.8 Workspace Layouts

The Workspace Layouts toolbar provides tools for quickly switching between pre-defined workspace layouts. It also provides tools for laying out your workspace then saving the resulting layout. The Workspace Layouts toolbar can be docked to any main window edge or floated freely in a convenient location for quick access.

Access: [View](#) | [Toolbars](#) | [Workspace Layouts](#)

Use to:

- Save the current workspace layout
- Manage workspace layouts
- Apply a selected workspace layout
- Switch to Views, such as Element List, Model Search and Relationship Matrix
- Activate or select main windows, such as Project Browser, Team Review, Element Properties and Execution Analyzer
- Toggle visibility of toolbars

Reference:



Action	Usage	Shortcut	See also
Save As	Save the current workspace layout to file		
Manage Layouts	Opens the Workspace Layouts dialog		
Select Layout	Select and apply a saved workspace layout		
Views	From a drop-down menu, choose one of several views to display <ul style="list-style-type: none"> • Element list • Model Search • Relationship Matrix • Auditing • Web Browser 		
Windows	Choose from a drop-down menu, one of many main windows to display or select. Windows include; <ul style="list-style-type: none"> • Project Browser • Team Review • Element Properties • Execution Analyzer 		
Toolbars	Toggle visibility of toolbars		
Fullscreen	Hide all docked windows and view the current view in full screen		

Notes:

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Workspace Layouts** menu option

2.5.5.9 Debug & Record Toolbar

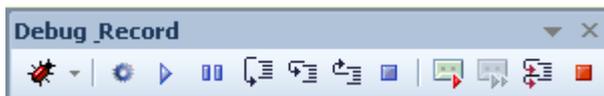
The Debug & Record toolbar provides tools for debugging program source code. The Debug & Record toolbar can be docked to any main window edge or floated freely in a convenient location for quick access.

Access: **View | Toolbars | Debug & Record**

Use to:

- Attach to running processes
- Run, pause or terminate program execution
- Step over, step into or step out of program statements
- Record program execution

Reference:



Action	Usage	Shortcut	See also
Drop-down Command List	Show drop-down menu containing commands for running and managing Analyzer scripts		
Attach to Process	Attach to a running process		
Debug	Begin or continue execution of the current program	F6	
Pause-Resume	Pause or resume program execution		
Step Over	Step over the next statement	Alt + F6	
Step In	Step into the next statement	Shift + F6	
Step Out	Step out of the current method	Ctrl + F6	
Debug Stop	Terminate execution of the current program	Ctrl + Alt + F6	
Manual Record	Manually step record the current thread		Recording Sequence Diagrams <small>1637</small>
Auto Record	Auto record an execution trace for the current thread		
Step Through	<p>Step through</p> <p>This button is available when manually recording steps; it attempts to step into a function on the line of code</p> <p>If a function call is made, the function call is captured and the debugger steps back out of the function to either the same line or the next</p>	Alt + ;	

Stop Recording	End recording of program execution		
-----------------------	------------------------------------	--	--

Notes:

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Diagram** menu option

2.5.5.10 Status Bar

The Status bar displays at the bottom of the Enterprise Architect workspace. It provides feedback on current operations and other status information, and enables you to manipulate the scale of a diagram.

Access: **View | Toolbars | Status Bar**

Use to:

- Identify the type and name of the currently selected element in a diagram (or the status of a Model Search)
- Identify the name of the currently-selected feature, if one is selected
- View the current coordinates of the top left corner of the selected element, and its width and height
- Adjust the scale of the current diagram by up to 50%, through use of a zoom slider
- Check the status of (**Caps Lock**), (**Num Lock**), (**ScrL Lock**) and the WAN Optimizer (bold indicates 'in use', pale indicates 'off')
- Check, by the presence of a triangle in the bottom right corner, that the screen is not maximized; you can drag the screen corner to increase the size of the window

Reference:

Action	Usage	Shortcut	See also
Right-click on Status Bar	Configure the status bar; open a context menu that enables you to hide or show the element name, element coordinates, zoom slider or status indicators		
Zoom Out	Zoom out on the current diagram		
Adjust Zoom	Adjust the scale of the current diagram by moving the slider		
Zoom In	Zoom in on the current diagram		

Notes:

- This facility has no impact on other users who might view the diagram
- The facility has the same function as the **Scale view by** field on the [Diagram Appearance](#) ⁴²⁷ page of the Options dialog; changes in the 'zoomed' display scale of a diagram update this field and are applied to any other diagrams that you open
- The facility also has no impact any other diagram Zoom facility in Enterprise Architect
- You can hide or show the toolbar by clicking on the **View | Toolbars | Status Bar** menu option

2.5.6 Customization

You can configure various settings using the Options dialog, which you display by selecting the **Tools | Options** menu option. In addition, there are several options to change the overall look and feel of Enterprise Architect in the **View | Visual Style** submenu. Those settings and options are explored in this topic.

On occasion, you might want to use Enterprise Architect for two distinct types of operation at the same time. You can do this by adding the following command line argument when you run Enterprise Architect:

```
/regkey: <regkeyname>
```

This stores registry settings - such as window layouts - to a different path in the registry.

Learn More:

- [Workspace Layouts](#) ^[115]
- [Local Options](#) ^[427]
- [Visual Styles](#) ^[125]

2.5.6.1 The Customize Dialog

The Customize dialog enables you to customize Enterprise Architect:

Access:

- Select the **Tools | Customize** option, or
- At the far right of any toolbar, click on the drop-down arrow and on the **Add or Remove buttons** option, then select the **Customize** option.

Topic	Link
Commands	Commands ^[115]
Toolbars	Toolbars ^[119]
Tools	Tools ^[120]
Keyboard Keystrokes	Keyboard Keystrokes ^[124]
Menus	Menus ^[125]
Options.	Options ^[125]

If a documented toolbar icon, keyboard combination or menu option does not appear to be available, select the appropriate tab and click on the **Reset** or **Reset All** button. This restores the toolbar, menu or key settings to the defaults.

However: Be aware that this also removes any customized icons, options or combinations you might have set, because it is possible that the customization itself has displaced or affected the default setting.

2.5.6.1.1 Customize Commands

The Customize dialog Commands tab provides access to many of Enterprise Architect's functions, enabling you to place them into a toolbar.

To add a command to a toolbar, click on the category in the Categories: panel and select the command from the list for that category in the Commands: panel. Drag the command on top of the toolbar to add it to.

If you right-click on the command icon in the toolbar while the Customize dialog is open, a context-sensitive menu displays. This menu offers options for deleting commands from a toolbar, and for changing the appearance of commands.

To remove a command from the toolbar, right-click on the command graphic or text and select the **Delete** menu option.

To change the appearance of a command graphic, right-click on the command graphic or text and select the **Button Appearance** context menu option. The Button Appearance dialog displays, which you can use to add graphical icons to commands that do not have them by default.

Some commands do not come with a convenient icon, which results in an empty toolbar button. Either avoid placing these commands on toolbars or use the context-sensitive menu to select an appropriate icon for the command.

Read the [Create a New Toolbar and Populate it with Commands](#) procedure of the *Customize Toolbars* topic.

2.5.6.1.2 Customize Toolbars

The Toolbars tab on the Customize dialog enables you to:

- Hide or show toolbars by selecting the appropriate checkbox
- Rename toolbars
- Create new toolbars
- Delete toolbars
- Modify toolbar contents by dragging commands onto a visible toolbar from the Commands tab or from another toolbar
- Reset a toolbar (or all toolbars) to the default contents and position, and
- Display text labels under the toolbar icons (perhaps temporarily, just to check what the icons do).

Create a New Toolbar and Populate it with Commands:

To create a new toolbar and populate it with commands, follow the steps below:

Step	Action	See also
1	Select the Tools Customize menu option The Customize dialog displays	
2	Click on the Toolbars tab	
3	Click on the New button The Toolbar Name dialog displays	
4	In the Toolbar Name field, type a name for your new toolbar and click on the OK button Your new toolbar is created	
5	To add commands from another toolbar to your new toolbar, see step 8 To add menu commands to your toolbar, click on the Commands tab This forces the new toolbar behind the Customize dialog, so you might have to drag the Customize dialog to the side to find your new toolbar	
6	Find the command to add to your toolbar in the Commands list The Categories list on the left represents the Enterprise Architect menu structure	

Step	Action	See also
	and the Commands list updates each time you click on a different category	
7	<p>Drag the selected command from the list into the new toolbar</p> <p>If you selected the Show text labels checkbox, your toolbar should now look like this:</p>  <p>If you did not select the Show text labels checkbox, your toolbar should look like this:</p> 	
8	<p>If the command you require is on another toolbar, display that toolbar and drag it to position it near your new toolbar</p> <p>Press (Ctrl) and drag the required icon for the command from the existing toolbar onto your new toolbar</p>	

Notes:

- You can select the **Show text labels** checkbox to display textual descriptions of toolbar items
- You can add as many commands to your toolbar as required; your new toolbar behaves the same way as other toolbars - you can position it next to the other toolbars at the top of the application workspace, dock it to the side of the workspace or close it
- On the Customize dialog, you can customize all the listed toolbars except for the Format Tool toolbar (which is actually tied to displayed diagrams)

Learn More:

- [Customize Commands](#)^[118]

2.5.6.1.3 Custom Tools

The Tools tab on the Customize dialog provides a means of extending the power of the Enterprise Architect desktop.

From this toolbar you can configure custom tools and make them accessible from the **Main Menu**. You can create menu options that hyperlink to different applications, compilers, batch scripts, automation scripts, URLs or documentation.

How to:

To add and configure custom tools, follow the steps below

Step	Action	See also
1	<p>Select the Tools Customize menu option</p> <p>The Customize dialog displays</p>	

Step	Action	See also
2	Click on the Tools tab	
3	Click on the New icon (left of the red X) A blank field displays in the Menu contents list	
4	Type in the name of the tool as it should appear in the menu	
5	In the Command field, type the name of the tool .exe file to use; the tool must be a valid filename	
6	Add any arguments required by the tool, and specify an initial directory if required	Opening External Tools ^[121] Passing Parameters to External Applications ^[123]
7	Close the Customize dialog Your tool should have now been added to the Tools menu	

2.5.6.1.3.1 Open External Tools

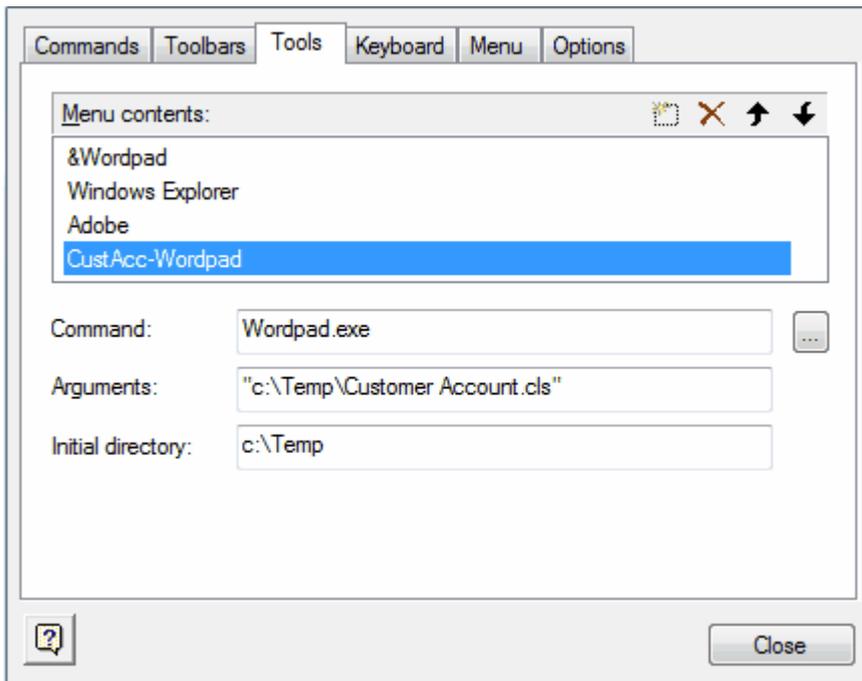
When configuring custom tools in Enterprise Architect, you can specify a file to be opened by the external application.

Select the **Tools | Customize** menu option. The **Customize** dialog displays; click on the **Tools** tab. Now you can:

- Specify a [custom tool](#) ^[120] (application) using the **Command** field
- Define a file to open or [parameters to pass](#) ^[123] to this application, using the **Arguments** field.

Example 1

This example opens the file `c:\Temp\Customer Account.xls` using Wordpad. If you save from within Wordpad the initial directory is `c:\Temp`.

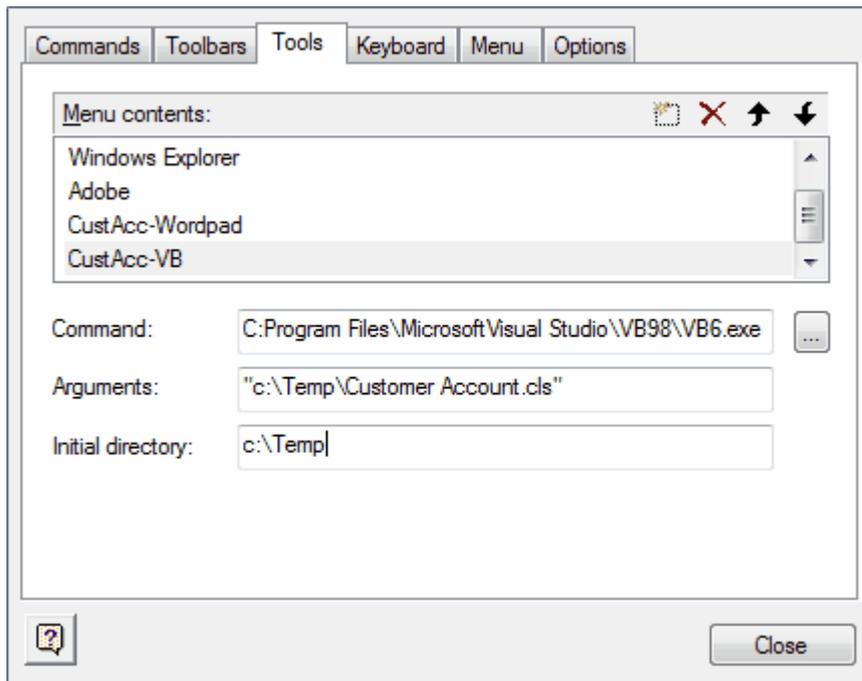


If there are any spaces in the paths in the **Command**, **Arguments** or Initial Directory fields, you must enclose the whole path in double quotes. For example:

" c : \ Temp \ Cust omer Account . cls " must have quotes but c:\Temp\CustomerAccount.cls does not have to have quotes.

Example 2

This example opens the file c : \ Temp \ Cust omer Account . cls using VB. As VB is not installed with the operating system, the whole file path for VB must be included in the **Command** field; you can select this using the (...) (Browse) button to locate the VB executable. If you save from within VB the initial directory is c : \ Temp .



2.5.6.1.3.2 Pass Parameters to Applications

When configuring custom tools in Enterprise Architect, you can pass parameters to the application.

Select the **Tools | Customize** menu option. The Customize dialog displays; click on the Tools tab. Now you can:

- Specify a custom tool (application) using the **Command** field
- Define a file to open or parameters to pass to this application using the **Arguments** field

The available parameters for passing information to external applications are:

Parameter	Description	Notes
\$f	Project Name	For example, C:\projects\EAexample.eap.
\$F	Calling Application (Enterprise Architect)	Enterprise Architect.
\$p	Current Package ID	For example, 144.
\$P	Package GUID	GUID for accessing this package.
\$d	Diagram ID	ID for accessing associated diagram.
\$D	Diagram GUID	GUID for accessing associated diagram.
\$e	Comma separated list of element IDs	All elements selected in the current diagram.
\$E	Comma separated list of element GUIDs	All elements selected in the current diagram.

Learn More:

- [Custom Tools](#) 

- [Opening External Tools](#)^[12]

For more information on using the Automation Interface, visit www.sparxsystems.com/AutIntVB.htm.

2.5.6.1.4 Customize Keyboard

The Keyboard tab on the Customize dialog enables you to configure shortcuts used to access main menu options. This is convenient for creating additional shortcut keys or for changing the current configuration to match your work habits or other applications.

Access: **Tools | Customize**

How To:

To modify a keyboard shortcut, follow the steps below:

Step	Action	See also
1	Select the Customize menu option The Customize dialog displays	
2	Click on the Keyboard tab In the Category field click on the drop-down arrow and select the menu containing the command to modify	
3	In the Command field, click on the drop-down arrow and select the command The current shortcut key (if any) for the command is displayed in the Current Keys field	
4	Move the cursor to the Press New Shortcut Key field and press the required shortcut key(s) for this command Press the actual keys to use; for example, to use (F5) press the (F5) key, don't type F then 5 The Assign button might be disabled if the selected keyboard shortcut is already used for another command; if this occurs the already existing command is shown and you must select a different shortcut key	
5	Once you have selected an available shortcut, click on the Assign button to apply the change	
6	Adding a keyboard shortcut to a command that already has a keyboard shortcut creates multiple shortcuts for a single command You can remove the previous shortcut by selecting it and clicking on the Remove button	

Notes:

- Modified shortcut keys are stored in the registry, so they only affect the current user
- You can revert to the default shortcut keys at any time, by clicking on the **Reset All** button

2.5.6.1.5 Customize Menu

The Menu tab on the Customize dialog enables you to customize the appearance of your menus.

Option	Description	See also
Application Frame Menus	Currently the Show Menus For feature is disabled as Enterprise Architect is not an MDI application	
Context Menus	Currently this feature is disabled	
Menu Animations	The following menu animations can be selected from the Menu animations drop-down list: <ul style="list-style-type: none"> • None • Unfold • Slide • Fade • [Default] 	
Menu Shadows	Menu shadows can be toggled on or off by selecting or clearing the Menu shadows checkbox	
Remove Menu Options	Some menu options might not be of relevance to you If you prefer not to display such options, follow the steps below <ol style="list-style-type: none"> 1. Whilst the Menu tab of the Customize dialog is displayed, click on the appropriate chain of options in the main menu bar to display the option to delete 2. Right-click on the option and select the Delete option from the context menu 	

Notes:

- When you perform a major upgrade of Enterprise Architect (such as from release 8.0 to 9.0) the menus are reset and deleted options are replaced.

2.5.6.1.6 Customize Options

The Options tab on the Customize dialog enables you to customize the appearance of toolbar items.

You can toggle the following options by selecting or clearing the checkboxes:

- **Show Screen Tips on toolbars**
- **Show shortcut keys in Screen Tips**
- **Use Large Icons.**

2.5.6.2 Visual Styles

You can configure the overall look and feel of Enterprise Architect to suit your working environment. Options include various Microsoft Office and Visual Studio styles, or themes.

Access: **View | Visual Style | Select Visual Style**

How to:

To reset the appearance of Enterprise Architect, follow the steps below:

Step	Action	See also
1	Select the Select Visual Style menu option The Application Look dialog displays	
2	Select the required style from the list If you select the Microsoft Office 2007 or Microsoft Office 2010 radio buttons, you can also select from a number of base-color options	
3	To try out styles, click on the Apply button To set the style and resume work, click on the OK button	

Notes:

- You can also enable customization of toolbars and menus, and animate auto-hidden windows

Learn More:

- [Customize Menu](#) ^[125]
- [Autohide Windows](#) ^[105]

2.5.6.3 Manage Workspace Layout

The **Manage Workspace Layouts** icon displays the **Workspace Layout** dialog, which lists the currently-available user-defined and system layouts.

In the **Existing Workspace Layout** panel at the bottom of the dialog, the **Based On:** field identifies the defined layout that the current workspace layout was derived from - you might have moved or closed windows since applying that layout. The highlighted (*Copy of existing Workspace Layout*) at the top of the **Workspace Layout** dialog is a capture of the workspace layout immediately before you opened the dialog.

Action	Description	See also
Change Layout	<p>You can now change the layout in the Existing Workspace Layout panel to:</p> <ul style="list-style-type: none"> The original layout (as identified by the Based On: field), discarding any changes you might have made The <i>Copy of existing Workspace Layout</i>, preparatory to saving the changes in a new named layout One of the other named layouts. <p>To change the layout in use, either:</p> <ul style="list-style-type: none"> Double-click on the required layout name Click on the layout name and click on the Apply button or OK button, or Right-click on the layout name and select the Apply context menu option. <p>When the layout in use changes, the layout name in the toolbar Workplace Layout Selection field also changes.</p>	
Copy Layout	<p>To copy a layout, either:</p> <ul style="list-style-type: none"> Change the layout in use to the required layout and click on the Save As button, or Right-click on the layout name and select the Save As 	

Action	Description	See also
	<p>context menu option.</p> <p>The Save Custom Workspace Layout dialog displays.</p> <p>In the Custom Workspace Layout Name field, type a name for the layout. Again, by selecting an existing name you can change an existing layout to something different. Click on the Save button.</p> <p>If you already have tailored windows or views that you want to include in your selected layout, select the Include active custom views checkbox.</p>	
Delete Layout	To delete a workspace layout, right-click on the layout name and select the Delete context menu option. Enterprise Architect prompts you to confirm or cancel the deletion.	

Workspace Layout Selection

Enterprise Architect provides a number of layouts of windows and toolbars to suit particular areas of work, such as Requirements Management, Code Engineering and Debugging. As described above, you can also add your own layouts to the drop-down list; these custom layout names have a preceding asterisk in the drop-down list for the field (the system-provided layouts are not marked with an asterisk).

To switch to another layout, click on the drop-down arrow and click on the required layout.

2.5.7 Other Windows

Most of the windows in Enterprise Architect have a specific, task oriented purpose. Two windows have broader functions.

Window	Detail	See also
System Output Window	Displays data that Enterprise Architect generates during each of a range of processes	The System Output Window ^[127]
Web Browser	Enables you to search for and use internet facilities within your Enterprise Architect work area.	The Web Browser ^[128]

2.5.7.1 The Output Window

The System Output window is used to display information generated by internal Enterprise Architect processes, or by Add-Ins and other third-party tools. This window can provide useful information during long-running processes and batch operations. Many validation processes within Enterprise Architect write out their results to this window.

Access:

- **View | System Output** (**Ctrl+Shift+6**)
- Element Context Menu
- Notes Window Context Menu

Use to:

- Validate information
- Progress information during launch of external processes

- Command line output from Build and Test
- Parse errors generated during import of various types of files
- Auditing Information *
- Re-docking the [Model Search](#)^[475] results into the Output window

How to:

Right-click on an item and select context menu options to:

- Copy the selected item to the clipboard
- Copy all items to the clipboard
- Save the output to an external file
- Clear the output from the window
- Rollback selected [Get All Latest](#)^[278] updates to a version controlled package

Drag selected items/text out of the System Output window and onto a diagram to quickly create a named element.

Double-click on model validation errors or parsing errors to display the source of the error.

Switch between tabs to view different categories of information from different tools.

Notes:

- The System Output window can also be used by [Add-Ins](#)^[1967], if they are configured to do so.
- * (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect) The [Audit History](#)^[307] tab of the Output window shows a history of changes to any element or connector selected from the Audit View, the Package Browser, the Diagram List, the Project Browser or the current diagram ([Auditing](#)^[298] must be turned on)

2.5.7.2 The Web Browser

The Web Browser displays as a tab of the central work area, like the Start Page, Model Search, Package Browser and Diagram View. It provides access within Enterprise Architect to internet facilities such as email, websites and search engines.

To access the Web Browser:

- Press (**Ctrl+Alt+W**), or
- Select the **View | More Project Tools | Internal Web Browser** menu option.

The Web Browser opens at the default home web site; you define the default home website, search engine and email exchange address on the General page of the Options dialog.

To access the:

- Email exchange server, click on the 'envelope' icon in the toolbar; the email login window displays
- Web search engine (such as *Google*), click on the 'spyglass' icon in the toolbar; the search engine screen displays
- Home web site, after displaying other web pages, click on the 'house' icon in the toolbar.

To go directly to another website or email server (your internet security permitting), in the **Address** field type or select the website http address and click on the **Go** button.

2.5.8 Keyboard Shortcuts

The table below lists the default keyboard shortcut functions within Enterprise Architect.

Notes:

- You can also display the key combinations on the Help Keyboard dialog (**Help | Keyboard Accelerator Map**; click on the drop-down arrow in the **Category** field and select the appropriate category)
- There are some additional shortcuts using the keyboard and mouse in combination; see the [Keyboard-Mouse Shortcuts](#)^[133] topic
- If necessary, you can change these keyboard shortcuts using the Keyboard tab of the [Customize](#)^[124] dialog

Action	Shortcut	Category
Create a new Enterprise Architect project	Ctrl + N	File
Open an Enterprise Architect project	Ctrl + O	File
Open Source File	Ctrl + Alt + O	File
Reload the current project ^[187]	Ctrl + Shift + F11	File
Print the active diagram	Ctrl + P	File
Undo Change	Ctrl + Z	Edit
Redo Change	Ctrl + Y	Edit
Add a single element to the clipboard list	Ctrl + Space	Edit
Paste element as metafile from clipboard	Ctrl + Shift + Insert	Edit
Paste element as new	Ctrl + Shift + V	Edit
Paste element(s) from the clipboard	Shift + Insert	Edit
Bookmark current element with red marker	Shift + Space	Edit
Delete selected element(s) in diagram	Delete or Ctrl + D	Edit
Delete selected element(s) from model (through diagram OR Project Browser)	Ctrl + Delete	Edit
Search for items in the project	Ctrl + F	Edit
Set focus to current window	Ctrl + Shift + 0	Window
Autohide the current window	Ctrl + Shift + F4	Window
Hide the current window	Ctrl + F4	Window
View Project Browser	Alt + 0	View
View Properties window	Alt + 1	View
View Project Information window	Alt + 2	View
View Testing window	Alt + 3	View
View Maintenance window	Alt + 4	View
Display Toolbox	Alt + 5	View
View Resources window	Alt + 6	View
View Source Code window	Alt + 7	View
View Debug Workbench	Alt + 8	View

Action	Shortcut	Category
View Notes window	Ctrl + Shift + 1	View
View Element Relationships window	Ctrl + Shift + 2	View
View Rules and Scenarios (Requirements and Constraints) window	Ctrl + Shift + 3	View
View Traceability window	Ctrl + Shift + 4	View
View Tagged Values window	Ctrl + Shift + 6	View
View Project Management window	Ctrl + Shift + 7	View
View Output window	Ctrl + Shift + 8	View
View Learning Center	Ctrl + Shift + 9	View
View Pan & Zoom Window	Ctrl + Shift + N	View
View Model Search	Ctrl + Alt + A	View
View Package Browser (or Diagram List)	Ctrl + Alt + R	View
Open Team Review	Ctrl + Alt + U	View
Display Web Browser	Ctrl + Alt + W	View
View Element Browser	Alt + 9	View
Add new package to project	Ctrl + W	Project
Add new diagram to package	Ctrl + Insert	Project
Add new element to package	Ctrl + M	Project
Create RTF documentation	F8	Project
Generate HTML Report	Shift + F8	Project
Generate Diagrams-only Report	Ctrl + Shift + F8	Project
Generate package source code	Ctrl + Alt + K	Project
Synchronize package contents	Ctrl + Alt + M	Project
Import source directory	Ctrl + Shift + U	Project
Manage Analyzer Scripts on Execution Analyzer window	Shift + F12	Project
Build	Ctrl + Shift + F12	Project
Test	Ctrl + Alt + T	Project
Run	Ctrl + Alt + N	Project
Deploy	Ctrl + Shift + Alt + F12	Project
Debug Run	F6	Project
Step Into	Shift + F6	Project
Step Over	Alt + F6	Project

Action	Shortcut	Category
Step Out	Ctrl + F6	Project
Debug Stop	Ctrl + Alt + F6	Project
Transform selected elements	Ctrl + H or Ctrl + Alt + F	Project
Transform current package	Ctrl + Shift + H	Project
Validate Selected	Ctrl + Alt + V	Project
Manage locks applied by current user	Ctrl + Shift + L	Project
Configure package control	Ctrl + Alt + P	Project
Import package from XMI	Ctrl + Alt + I	Project
Export package to XMI	Ctrl + Alt + E	Project
Import and export to CSV files	Ctrl + Alt + C	Project
Manage Baselines	Ctrl + Alt + B	Project
Diagram properties	F5	Diagram
Save	Ctrl + S	Diagram
Save image to file	Ctrl + T	Diagram
Save image to clipboard	Ctrl + B	Diagram
Visible Relations	Ctrl + Shift + I	Diagram
Locate in Project Browser	Shift + Alt + G	Diagram
Repeat last element	Shift + F3 or Ctrl + click	Diagram
Repeat last connector	F3	Diagram
Element Properties	Alt + Enter	Element
Add Tagged Value	Ctrl + Shift + T	Element
Linked Document	Ctrl + Alt + D	Element
Display Attribute Properties dialog	F9	Element
Display Operation Properties dialog	F10	Element
Space elements evenly horizontally	Alt + -	Element
Space elements evenly vertically	Alt + =	Element
Add attribute	Ctrl + Shift + F9	Element
Add operation	Ctrl + Shift + F10	Element
Add other type	Ctrl + F11	Element
Auto-size selected elements	Alt + Z	Element
Generate code from element	Ctrl + G or F11	Element

Action	Shortcut	Category
Move element by increments	Shift + ↑, ↓, →, ←	Element
Resize selected element	Ctrl + ↑, ↓, ←, →	Element
Align bottom edges of selected elements	Ctrl + Alt + Down	Element
Align top edges of selected elements	Ctrl + Alt + Up	Element
Align selected elements on left boundaries	Ctrl + Alt + Left	Element
Align selected elements on right boundaries	Ctrl + Alt + Right	Element
Configure element default appearance	Ctrl + Shift + E or F4	Element
Edit selected	F2	Element
Manage embedded elements	Ctrl + Shift + B	Element
Insert new feature after current selection	Insert	Element
Locate in browser	Alt + G	Element
New element	Ctrl + M	Element
View source code in default editor	Ctrl + E or F12	Element
Operation	F10	Element
Override inherited features	Ctrl + Shift + O	Element
Configure element properties	Alt + Enter	Element
Select alternative image	Ctrl + Shift + W	Element
Specify which element features are visible on a diagram	Ctrl + Shift + Y	Element
Set element parent or implement interface(s)	Ctrl + I	Element
Set references to other elements and diagrams	Ctrl + J	Element
Create Workbench Instance	Ctrl + Shift + J	Element
Locate diagrams where element is used	Ctrl + U	Element
View Properties dialog	Enter	Element
Check project data integrity	Shift + F9	Tools
Configure system options	Ctrl + F9	Tools
Spell check current package	Ctrl + Shift + F7	Tools
Spell check model	Ctrl + F7	Tools
Spell check Notes text	F7	Object notes
Edit code generation templates	Ctrl + Shift + P	Settings
Edit transformation templates	Ctrl + Alt + H	Settings
Make text bullet list item	Ctrl + . (full stop)	Object notes

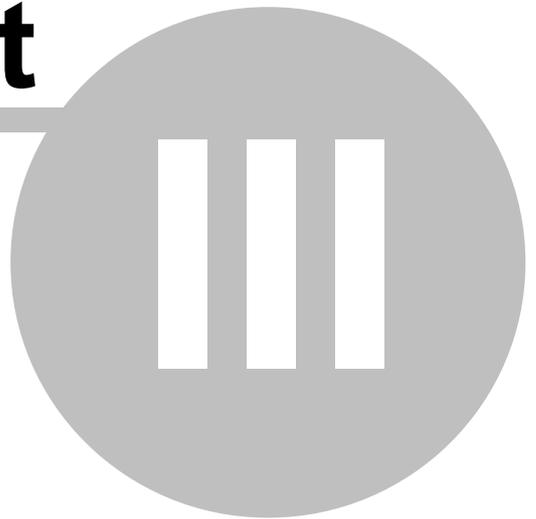
Action	Shortcut	Category
Make text numbered list item	Ctrl + 1	Object notes
Make text bold	Ctrl + B	Object notes
Make text italic	Ctrl + I	Object notes
Make text underlined	Ctrl + U	Object notes
Insert date and time in Notes text	F5	Object notes
Select line of text in Notes text	F8	Object notes
Copy text	Ctrl + C	Everywhere
Paste text	Ctrl + V	Everywhere
Cut text, or element in diagram	Ctrl + X	Everywhere

2.5.8.1 Keyboard-Mouse Shortcuts

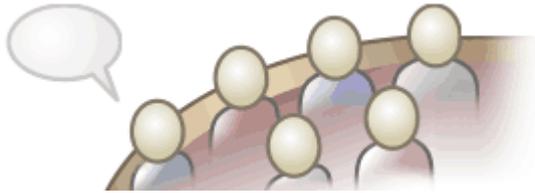
The following table lists certain operations you can perform quickly using a combination of keyboard keys and the computer mouse.

Action	Shortcut
Create element of same type as last created on diagram	Ctrl + click
Display element Properties dialog for element in scenario Context References tab	Ctrl + click
Select a number of individual objects for an operation	Ctrl + click each object
Select a range of objects for an operation	Shift + click the first and last object in the range
Move elements along horizontal axis or vertical axis	Shift + hold left mouse button and drag
Create bend in and change direction of connector line	Shift + hold left mouse button and drag
Move elements in any direction, including diagonally	Alt + hold left mouse button and drag
Zoom in or zoom out of diagram	Ctrl + mouse scroll wheel
Pan diagram horizontally or vertically	Ctrl + Shift + hold left mouse button and drag

Part



3 Projects and Teams

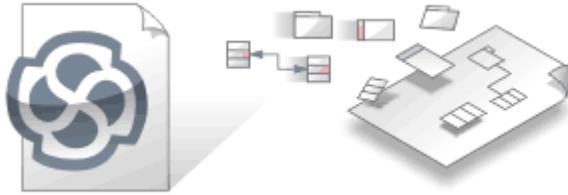


Enterprise Architect helps you to create projects for development under a [range of work conditions](#)^[136], from single user/local access through to multiple-role teams working in a distributed environment. You both protect and manage the model data itself, and communicate information on the data in the form of documentation and reports, using facilities such as those listed below.

Topics:

Topic	Link
Creating a project in a simple, file-based repository (.EAP file)	File-Based Repositories ^[145]
Creating the project in one of a range of DBMS repositories - these provide for larger models with more concurrently connected users (Corporate and extended editions)	Server-Based Repositories ^[148]
Tools for enabling team or multiple-user development in the project	Team Development ^[185]
Tools for managing change within the model	Change Management ^[241]
A variety of tools for managing modeling and project activities	Project Management ^[346]
Maintaining the integrity of the project data	Project Maintenance ^[414]
Sharing the reference data used across the project, between models and between projects	Sharing Reference Data ^[236] Reference Data ^[772]
Enabling each user to configure their personal preferences for how project tools display and behave on their workstation	Local Options ^[421]
You can have recorded discussion and communication of decisions using the Team Review	Team Review Tools ^[216]
Managing work and resources allocated to tasks	Project Task Allocation ^[366]
Communicating and collaborating on model development through the Model Mail facilities	Personal Information ^[371]
Tracking important project events, resource issues and allocations through the Project Calendar	Project Calendar ^[390]
Documenting your model through RTF or HTML reports, which you can tailor to your purposes	Reporting ^[1724]

3.1 Introduction



An Enterprise Architect project is stored in a data repository. In Enterprise Architect Desktop and Professional editions, you work with a single file having a .EAP extension. In Enterprise Architect Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions you can also use a suitable DBMS database for project files.

Topics:

Topic	Detail	See also
Project Files	<p>.EAP Files</p> <p>In Enterprise Architect Desktop and Professional editions, a single file with a .EAP extension is used to store projects. A .EAP file is a Microsoft JET database, so you can also open it using MS Access 97, 2000 or 2003, or any other reporting tool that can work with JET databases.</p> <p>DBMS Repositories</p> <p>In Enterprise Architect Corporate Business and Software Engineering, Systems Engineering and Ultimate editions, you can use a suitable DBMS database for project files. DBMS project files have the same logical structure as .EAP files, but must be connected to using ADO/ODBC. See Connect to a Data Repository, below</p>	<p>File Based Repositories ^[145]</p> <p>Server Based Repositories ^[145]</p>
Creating Project Files	<p>On creating a new project, the Model Wizard enables you to create a model containing various Model Packages.</p> <p>You can also add Model Packages to a project from the Project Browser by:</p> <ul style="list-style-type: none"> • Right-clicking on an existing model and selecting the New Model or Add a New Model using Wizard context menu options • Right-clicking on a package and selecting the Add Add a New Model using Wizard context menu option • Clicking on an existing model, pressing (Insert) and selecting the New Model or Add a New Model using Wizard context menu options • Clicking on a package, pressing (Insert) and selecting the Add a New Model using Wizard context menu option 	<p>File Based Repositories ^[145]</p> <p>Model Wizard ^[518]</p>
Opening Existing Projects	<p>There are various ways to open a project in Enterprise Architect. New users are advised to explore the EAExample file supplied with Enterprise Architect.</p>	<p>Open a project ^[138]</p>
Connect to a Data Repository	<p>Enterprise Architect enables you to connect to any of the following data repositories:</p>	<p>Connect to a Data Repository ^[161]</p>

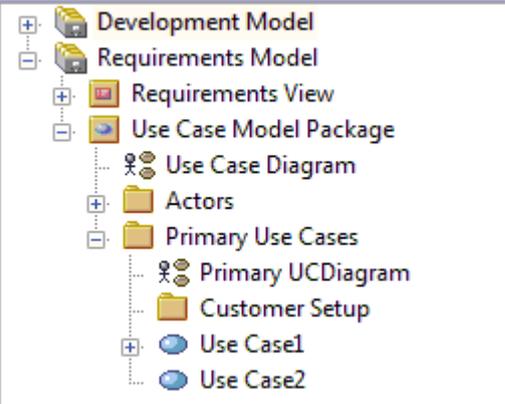
Topic	Detail	See also
	<ul style="list-style-type: none"> • MS Access 97, 2000 and 2003 (in all editions - .EAP files are stored in Microsoft JET databases) • Access 2007 • SQL Server 2000, 2005 and 2008 • MySQL • Oracle 9i, 10g or 11g • PostgreSQL • MSDE • Adaptive Server Anywhere • Progress OpenEdge <p>To create a new data repository, you must first create a new database with the DBMS management software, then run supplied scripts to create the logical structure. You should then use Enterprise Architect data transfer functions to move a project from a .EAP or DBMS model into the new project.</p>	

3.1.1 Projects Defined

An Enterprise Architect Project is a repository for storing, manipulating and managing one or more Models. A single repository can contain many models, and a repository can be either file based or hosted in a larger DBMS system.

Guide:

Topic	Detail	See also
Project	A project can contain a single model, or a number of models, each of which defines a particular system or process. A model contains the diagrams, elements, relationships and associated metadata that define the structure and function of the system or process. These components are organized into a hierarchy of packages, which help to group and manage related components.	Open a Project ^[138] Model Shortcuts ^[140] Project Management ^[346] Team Development ^[185]
Model Package	Different aspects of the process or system - or their development - are defined by Model Packages, which you generate from templates specifically structured to support the aspects that the Model Packages represent, such as requirements or deployment. You can generate these templated packages at any level of the hierarchy, but as they are created with their own content they are more useful at the top levels.	Model Wizard ^[518] Model Templates ^[519]
View	The top-level packages in a model can also be Views, which represent partitions of the model that you define yourself. You can start with standard Views such as Class or Component, or create whatever partitions are appropriate to your model.	Views ^[530]

Topic	Detail	See also
Example Project Structure	 <p>Each View or Model Package contains packages. Use Case Model Package contains:</p> <ul style="list-style-type: none"> • Actors and • Primary Use Cases. <p>It also contains the diagram Use Case Diagram, which could be an overview of the package structure or function. Each package itself can contain one or more diagrams, one or more packages, and several elements. The Primary Use Cases package contains the:</p> <ul style="list-style-type: none"> • Primary UCDiagram • Customer Setup package • Use Case 1 element • Use Case 2 element <p>Each subordinate package also contains diagrams, elements and (if necessary) further packages. The elements are related by connectors created in the diagrams, and each element and connector has properties, attributes, operations and extensions defined in the respective Properties dialogs.</p>	
Storage	<p>A project can be a .EAP file in an MS Access database or (in the Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions) a structure of files in a database management system such as MySQL or Oracle.</p>	<p>File Based Repositories ^[145]</p> <p>Server Based Repositories ^[148]</p>

3.1.2 Open a Project

An Enterprise Architect project is used for storing and managing the components of one or more UML models. The Desktop and Professional versions of Enterprise Architect use an MS JET database as the model repository. If you are using the Corporate edition (or above), you can also use DBMSs such as Oracle and MySQL to host the model repository.

Access: **File | Open Project**

Use to:

- Create a new Enterprise Architect project file (the model repository is created as an MS Jet database file)
- Browse for an existing project file to open
- Specify connection details for opening a project hosted on a DBMS

- Select a recent project to open
- Remove items from the list of recent projects

Reference:

Field	Usage	See also
New Project...	Create a new project, as an EAP file	File Based Repositories [145]
Browse for Project	Open a file browser dialog, to select an existing EAP file; the EAP file can be a project file or a shortcut to a project hosted on a DBMS	Model Shortcuts [140]
Connect to Server	Specify connection details for opening an Enterprise Architect project that is hosted on a DBMS If you double-click on the option, the Windows Data Link Properties dialog displays; start to define the connection to the project's server-based repository If you click on the drop-down arrow and select the Connection Wizard option, the Data Link Properties dialog again displays If you click on the drop-down arrow and select the Connection String option, the Connection String dialog displays; type in or paste the connection string and click on the OK button to connect directly to the project	Server-Based Repositories [148]
Recent Projects	A list of the ten most recently opened projects This list is also displayed: <ul style="list-style-type: none"> • On the Start Page, under the Recent heading • As a drop-down menu under the Open Project button on the Default Tools toolbar • As part of the File menu On the Open Enterprise Architect Project dialog, each project name is shown with its access path You can edit the access path by right-clicking on it and selecting the Edit Connection String context menu option; make the changes on the Connection String dialog	
Remove Selection from List	Remove the currently selected project from the Recent Projects list	
Show this Dialog at Startup	Select whether or not to show the Open Enterprise Architect Project dialog upon starting Enterprise Architect	
Open	Open the currently selected project in the Recent Project list	
Cancel	Close this dialog without any further action	
Help	Display this Help topic	

Notes:

- Use of a DBMS for hosting the model repository is only available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- Enterprise Architect supports the following DBMS products for hosting model repositories; [SQL Server](#) ^[150], [MySQL](#) ^[149], [Oracle 9i, 10g or 11g](#) ^[151], [Postgre SQL](#) ^[167], [ASA](#) ^[169], [MSDE Server](#) ^[170] and [Progress OpenEdge](#) ^[170]

Learn More:

- [Projects Defined](#) ^[137]
- [File Based Repositories](#) ^[145]
- [DBMS Repositories](#) ^[148]
- [Team Development](#) ^[185]

3.1.3 Project Shortcuts

Enterprise Architect enables you to create a desktop shortcut (or *proxy* file) to a real Enterprise Architect project (DBMS or file-based). Each shortcut is a file containing the connection string for the model.

However, the shortcut also defines *views* that Enterprise Architect should open when it opens the model, as outlined below.

Topics:

View	Detail	See also
Diagrams	Define one or more specific diagram(s) to be loaded on Open	
Model Search	Open the Model Search with a specific text string and search type For searches operating on the current tree selection, a diagram in the target package must be opened first. If you use a custom SQL search, the SQL must include <i>ea_guid AS CLASSGUID</i> and the object type You cannot define more than one Model Search to open	Model Search ^[475]
The Relationship Matrix with a saved profile	Open the Relationship Matrix with a saved profile You cannot define more than one Relationship Matrix profile to open	Relationship Matrix ^[496]
The default Team Review	Open the default Team Review document You cannot define more than one Team Review to open	Team Review Tools ^[216]
An example shortcut	You might create a shortcut to open, in sequence: <ul style="list-style-type: none"> • A Development module • The Model Search for a simple search on the term <i>Issue</i> • The module <i>Issues</i> diagram • The module <i>Changes</i> diagram <p>Enterprise Architect opens the appropriate windows in the sequence in which you list the options, displaying the last view in the list</p> <p>In this example, the project opens with the Enterprise Architect work area showing the two diagram tabs and the Model Search tab, and with the <i>Changes</i> diagram displayed in the</p>	

View	Detail	See also
	Diagram View	

Notes:

- If specified, the shortcut views override any default diagram defined for the model or current user
- A shortcut does not affect the original Enterprise Architect .exe file or icon, or any other shortcut you might have defined; you can use all of these independently
- When you use a shortcut to access a project that you have recently opened in Enterprise Architect, the **Recent** list on the Enterprise Architect Start Page has two entries for the project - one created when you opened the project in Enterprise Architect and one created when you used the desktop shortcut
- If you are using a database repository other than MS Access 97, 2000 or 2003, you can configure the shortcut to encrypt the password used to set up the connection between Enterprise Architect and the repository; the Enterprise Architect user does not have the real password, thereby preventing them from accessing the repository using other tools such as Query Analyzer or SQLPlus

Learn More:

- [Create Project Shortcut](#)^[141]
- [Capture Current Work Environment](#)^[142]
- [Encrypt Repository Password](#)^[143]
- [Default Diagram](#)^[83]
- [Creating Search Filters](#)^[484]

3.1.3.1 Create Project Shortcut

You can create a shortcut to an Enterprise Architect project (either a DBMS project or file based project). The shortcut can specify additional windows and diagrams to open up automatically every time the shortcut is run.

Shortcuts are stored with a .EAP extension, but are actually small text files that tell Enterprise Architect what project to open and what initial views and windows to display.

Access: **File | Save Project Shortcut**

Use To:

- Specify a working environment in advance, for other users

How to:

To create a project shortcut, follow the steps below:

Step	Action	See also
1	Open Enterprise Architect	
2	Open the required project	
3	Select the Save Project Shortcut menu option The Save Project Shortcut dialog displays	
4	Click on the (...) (Browse) button at the end of the Target File field	

Step	Action	See also
	The Save Project As dialog displays	
5	Browse for the appropriate file location and, in the File name field, type an appropriate filename All shortcuts are .EAP files, regardless of whether the model itself is a .EAP file or a DBMS model	
6	Click on the Save button to return to the Save Project Shortcut dialog	
7	Click on the Add Other button and select the required option to define: <ul style="list-style-type: none"> • A diagram to open • A Relationship Matrix profile to open • The Team Review • A Model Search to perform 	
8	The appropriate browser or dialog displays to define the view to display; enter the details and click on the OK button The view is added to the Actions when model is opened field; the entry is automatically selected, with a tick in the checkbox	
9	Repeat steps 7 and 8 for as many additional views as you require	
10	Review the items in the Actions when model is opened field <ul style="list-style-type: none"> • If you decide not to have an item in the shortcut, deselect its checkbox • If you want to clear all selected items, click on the Include None button Unselected entries are deleted when you save the shortcut	
11	If you decide to change the sequence and/or make a different view display first in the Diagram View: <ul style="list-style-type: none"> • Click on the appropriate entry • Click on the 'Up Hand' or 'Down Hand' buttons 	
12	Click on the OK button to save the shortcut	

Notes:

- When you subsequently open the Save Project Shortcut dialog, it lists the currently-opened views in the order in which they were opened; you can add further views or remove them from the shortcut

Learn More:

- [Capture Current Work Environment](#)^[142]
- [Encrypt Repository Password](#)^[143]

3.1.3.2 Capture Current Work Environment

You can capture the current Enterprise Architect work environment in your shortcut.

Access: **File | Save Project Shortcut**

Use To:

- Capture the current Enterprise Architect work environment, to access the model at exactly the same

point in exactly the same environment when you resume work

How to:

To capture your current work environment, follow the steps below:

Step	Action	See also
1	Open Enterprise Architect	
2	Open the required project and work in it	
3	At the point at which you decide to capture your work environment in a shortcut, ensure that: <ul style="list-style-type: none"> You have opened all diagrams you require If necessary, you have opened the Team Review, Model Search (with appropriate search term and type) and/or Relationship Matrix (at the appropriate profile) The view you want to resume work on is the last one opened 	
4	Select the Save Project Shortcut menu option The Save Project Shortcut dialog displays, showing a list of actions derived from the views you currently have open	
5	If you accessed Enterprise Architect via a shortcut, the Target File field displays the file location of that shortcut Otherwise, click on the (...) (Browse) button at the end of the Target File field The Save Project As dialog displays	
6	Browse for the appropriate file location and, in the File name field, type an appropriate filename All shortcuts are .EAP files, regardless of whether the model itself is a .EAP file or a DBMS model	
7	Click on the Save button to return to the Save Project Shortcut dialog	
8	In the Actions when model is opened field, click on the Include All button	
9	Click on the OK button to save the shortcut	

Learn More:

- [Encrypt Repository Password](#)^[143]

3.1.3.3 Encrypt Repository Password

If your model is developed on a DBMS repository, the Save Project Shortcut dialog has an **Encrypt Connection String** check box.

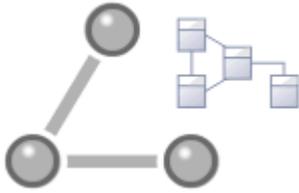
You can create the shortcut actions and, if necessary, select the checkbox to encrypt the database connection string.

You distribute the shortcut file to the database users who are to access the model. The users then have an encrypted string that prevents them from directly accessing the database using other tools.

Learn More:

- [Create Project Shortcut](#)¹⁴¹

3.2 File Based Repositories



In the Desktop and Professional editions of Enterprise Architect, you create Enterprise Architect projects as .eap files. If you prefer, you can also create your projects as .eap files in the Corporate and suite editions of Enterprise Architect.

A default project file (*EABase.EAP*) is supplied in the Enterprise Architect installer and is automatically stored in your Enterprise Architect installation directory; you can copy this as the base for your own projects. You can also create your own base projects, and copy existing operational projects.

Topic	Detail	See also
Create .EAP Project Files	<p>Select one of:</p> <ul style="list-style-type: none"> • File New Project option • New Project option on the Open Enterprise Architect Project dialog • Create a New Project option on the Start Page <p>All of these options display the New Project dialog; select a directory and enter a file name for your project</p> <p>Once the project has been saved, the Select Model(s) dialog displays, which makes a selection of Model Packages available; select the Model Packages to include</p> <p>Enterprise Architect adds a model containing the selected Model Packages to the Project Browser</p>	Model Wizard ^[518]
Default EABase.EAP Installation directories	<p>The default installation directories, depending on which version you have installed, are:</p> <ul style="list-style-type: none"> • Registered version: C:\Program Files\Sparx Systems\EA • Trial version: C:\Program Files\Sparx Systems\EA Trial • Lite version: C:\Program Files\Sparx Systems\EA Lite <p>Having copied the base project as a template for your own project, you can rename it</p>	Copy a Base Project ^[146] Rename a Project ^[418]
Configure Project	<p>Having created your project, you can set a range of project parameters to define defaults, tailor the project to particular coding languages, and ensure consistent development and use of the project</p>	The Settings Menu ^[98] Defaults and User Settings ^[118]
Create Custom Templates	<p>You can customize any Enterprise Architect project as a template project with company standards, tutorials, frameworks and any other common piece of modeling already in-built; with careful planning you can save yourself many hours of work at project start-up</p>	Incorporate Model Templates ^[108]
Copy Existing Project	<p>A Base project contains templates and reference data to enable you to develop your own project quickly</p> <p>You can also copy an operational .EAP project file to enable</p>	Copy Existing Projects ^[147]

Topic	Detail	See also
	separate development by team members, or to create an evaluation or distribution version of the project	

Notes:

- You can also add Model Packages to a project using the **New Model From Pattern** icon in the Project Browser toolbar

3.2.1 Copy a Base Project

You can copy an existing project as a new project. When using an existing project as the template for a new one, it is important to use this method, rather than simply copying the file using Windows Explorer. This process resets all the unique identifiers for packages and elements, so that your new project is truly unique - otherwise it is simply an exact copy of the source project.

Access: [Start Page: Copy a Base Project](#)

Use to:

- Create a project already largely set up, from a template project containing company standards, tutorials, frameworks and any other common piece of modeling
- Ensure all unique ID's (GUIDS) in a model are reset when starting a new model from an existing master

How to:

Step	Action	See also
1	On the Enterprise Architect Start Page, click on the Copy a Base Project option The Create New Enterprise Architect Project dialog displays	Start Page ^[70]
2	In the Model Project field, select the project that is the base template The field defaults to EABase.eap, the default base project provided in the Enterprise Architect installer, but you can click on the Browse button and locate any other custom project file to use	Incorporate Model Templates ^[1087]
3	In the New Project field, click on the Browse button and select the file path for saving your project If this is to be a shared project, store the file on a shared network resource such as a Network Server or Workgroup Server	
4	To replace all GUIDs from the source model with fresh GUIDs in the new model, select the Reset New Project GUIDs checkbox However, if the new project is based on one that is already under version control, we recommended that you deselect this checkbox to prevent duplication of packages when the <i>Get Latest</i> facility is used	Package Version Control Menu ^[278]
5	Click on the Create Project button to create your project	

3.2.2 Copy Existing Project

You can copy an operational .EAP project file to a new file location under a new name.

Access: **File | Save Project Copy**

Use to:

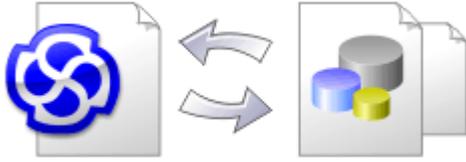
- Provide separate copies for individual team members
- Create an evaluation or distribution version of the project

How to:

To create a copy of a project, follow the steps below:

Step	Action	See also
1	Select the Save Project Copy menu option The Save Enterprise Architect Project Copy dialog displays	
2	In the Target Project field, type the new project filename You can either type in the file path as well, or click on the Browse button and browse for the required file path	
3	If this copy is to be used for a different project, select the Reset New Project GUIDs checkbox This ensures that all components of the new project have different IDs to their counterparts in the source project	
4	Click on the Save As button to save the new project	

3.3 Server Based Repositories



If you purchase the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions, you can upsize the .eap templates (such as EABase.eap) or any existing .eap project to a DBMS data repository.

A DBMS repository enables you to support larger models with more concurrently connected users.

Topic	Detail	See also
Supported DBMSs	<ul style="list-style-type: none"> • SQL Server 2000, 2005 or 2008 • MySQL 4 or 5 • PostgreSQL 7, 8 or 9 • Adaptive Server Anywhere 8 or 9, or SQL Anywhere 10, 11 or 12 • Access 2007 • Progress OpenEdge • MSDE or • Oracle 9i, 10g or 11g 	
Upsize Project to DBMS Repository	<p>To upsize the Enterprise Architect models (either existing or template) to use your selected DBMS:</p> <ol style="list-style-type: none"> 1. Install the DBMS software; how you do this is explained in your DBMS documentation and beyond the scope of the Enterprise Architect documentation 2. Create a repository and run a script supplied by Sparx Systems to create the required tables 3. Set up an ODBC driver to enable connection to the repository(not all DBMSs require this) 4. Open Enterprise Architect and perform a Project Integrity Check on the project .eap file (existing or template); this ensures the data is clean before uploading 5. In Enterprise Architect, open the Project Transfer dialog, connect to the .EAP file and the repository, and transfer the project data from the .EAP file to the DBMS repository <p>You cannot move a model from a source .EAP file of a version earlier than 3.5.0.</p>	<p>Create a Repository^[148]</p> <p>Sparx Systems Scripts</p> <p>Set Up an ODBC Driver^[154]</p> <p>Check Project Data Integrity^[414]</p> <p>Connect to a Data Repository^[161]</p> <p>Perform a Project Data Transfer^[343]</p>

3.3.1 Create a Repository

A repository hosted in a DBMS system requires some initial setup and configuration, and the actual steps required differ depending on the DBMS chosen. In general you must setup your database management system, configure a new database and then populate that database using the scripts provided by Sparx Systems. As a final step, you populate the new repository using a project transfer of the base model.

For more information and the steps involved for specific systems, follow the links below:

Topic	Link
MySQL	Create a MySQL Data Repository ^[149]
Access	Create An Access Data Repository ^[149]
SQL Server	Create a SQL Server Data Repository ^[150]
Oracle	Create an Oracle Data Repository ^[151]
PostgreSQL	Create a PostgreSQL Data Repository ^[151]
Adaptive Server Anywhere	Create an Adaptive Server Anywhere Data Repository ^[152]
MSDE Server	Create an MSDE Server Data Repository ^[152]
Progress OpenEdge	Create a Progress OpenEdge Data Repository ^[153]

Learn More:

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

3.3.1.1 Access 2007 Repository

The process of creating an Access 2007 repository is very simple. Using Access 2007, open the .EAP file and allow Access to convert it to a .ACCDB file. This forms the Access 2007 repository.

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- If you do not have Access 2007, you can download the Access Database Engine from the [Microsoft downloads site](#)
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

3.3.1.2 MySQL Repository

Creating a MySQL database is beyond the scope of this User Guide - refer to your MySQL documentation for assistance.

Having created your MySQL database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect.

You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

Third Party Tools:

If you are unfamiliar with MySQL and DBMS systems in general, you might want to consider a suitable front end tool. MySQL Administrator is one such tool, providing a convenient graphical user interface to enable the creation of databases, execution of scripts, backups and restores.

You might, therefore:

1. Run MySQL Administrator and create a new database
2. Run MySQL Query Browser, and open and execute the MySQL repository script

After creating a MySQL data repository in Enterprise Architect, you must set up the MySQL ODBC drivers.

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn More:

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [MySQL Workbench \(including MySQL Administrator\) download site](#)
- [MySQL ODBC Driver^{\[154\]}](#)

3.3.1.3 SQL Server Repository

Before creating a SQL Server data repository, you must have SQL Server and MDAC 2.6 or higher installed, and access permission to create a new database. Creating a SQL Server database is beyond the scope of this User Guide - refer to your SQL Server documentation for assistance.

Having created your SQL Server database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

How to:

If you are unfamiliar with SQL Server and DBMS systems in general, you might want to consider a suitable front end tool, such as SQL Enterprise Manager. The following example shows how to create the repository in SQL Enterprise Manager.

Step	Action
1	In SQL Enterprise Manager, locate the server on which to create your new Enterprise Architect model; for example: DBSERVER02\SQLEXPRESS
2	Right-click and choose the New Database context menu option
3	Enter a suitable name for the database. Set any file options as required Ensure that the database collation is set to the alphabet you use, such as Latin or Cyrillic, and case-insensitive
4	Click on the database to select it, then select the New Query menu option
5	In the Query window, use the Open File dialog to locate the supplied Enterprise Architect SQL Server Model script file
6	Click on the Open button. In the drop-down menu, check that you have selected the correct database to run the script in
7	Click on the Execute button; SQL Server executes the script, which creates the base model for an Enterprise Architect project

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

3.3.1.4 Oracle Data Repository

Before creating an Oracle data repository, you must have the appropriate version of Oracle (9i, 10g or 11g) and MDAC 2.6 or higher installed, and access permission to create a new database. Creating an Oracle database is beyond the scope of this User Guide - refer to your Oracle documentation for assistance.

Having created your Oracle database, use the SQL script provided by Sparx Systems to create the required table definitions and indexes for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

Third Party Tools:

If you are unfamiliar with Oracle and DBMS systems in general, you might want to consider a suitable front end tool. You could connect to the database and execute the scripts with a program such as Oracle SQL*Plus or SQL Plus Worksheet.

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn More:

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

3.3.1.5 PostgreSQL Repository

Creating a PostgreSQL database is beyond the scope of this User Guide - refer to your PostgreSQL documentation for assistance.

Having created your PostgreSQL database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

Third Party Tools:

If you are unfamiliar with PostgreSQL and DBMS systems in general, you might want to consider a suitable front end tool. One such tool is *pgAdminIII*, It provides a convenient graphical user interface to enable creation of databases, execution of scripts and restores.

After creating a PostgreSQL data repository in Enterprise Architect, you must set up the PostgreSQL and PostgreSQL ODBC drivers.

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn More:

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [pgAdminIII Download Site](#)
- [Set up a PostgreSQL ODBC Driver](#)^[155]

3.3.1.6 Adaptive Server Anywhere Repository

Creating an ASA database is beyond the scope of this User Guide - refer to your Sybase Adaptive Server Anywhere documentation for assistance.

Having created your ASA database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

Third Party Tools:

If you are unfamiliar with ASA and DBMS systems in general, you might want to consider a suitable front end tool. Sybase Central is one such tool, that can be installed along with the DBMS. It provides a convenient graphical user interface to enable creation of databases, execution of scripts and restores.

After creating an ASA data repository in Enterprise Architect, you must set up the ASA ODBC driver.

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn More:

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [Setup an Adaptive Server Anywhere ODBC Driver](#)^[155]

3.3.1.7 MSDE Server Repository

Before creating an MSDE data repository, you must have MSDE Server and MDAC 2.6 or higher installed. Creating an MSDE Server database is beyond the scope of this User Guide - refer to your MSDE Server documentation for assistance.

Having created your MSDE Server database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)

- Corporate edition Resources page (Trial users)

How to:

- Create the MSDE Server repository in the same way as a SQL Server repository

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn More:

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [Create an SQL Server Data Repository](#)^[150]

3.3.1.8 Progress OpenEdge Repository

Before creating a Progress OpenEdge data repository, you must have OpenEdge 10.0B03, 10.1B01 or later, and MDAC 2.6 or higher installed, and access permission to create a new database. Creating an OpenEdge database is beyond the scope of this User Guide - refer to your OpenEdge documentation for assistance.

Having created your OpenEdge database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

How to:

Step	Action
1	Run proenv from the OpenEdge menu: Start Programs OpenEdge proenv
2	Create the database: <code>proodb <database_name> empty.</code>
3	Start the database server: <code>proserve <database_name> -S <port_number></code>
4	Open Data Administration to add a user: <code>prowin32 -db <database_name> -S <port_number> -p _admin -rx</code>
5	Select Admin Security Edit User List .
6	Close Data Administration.
7	Open the SQL Explorer Tool, connect as <i>sysprogress</i>
8	Add a user: <code>create user 'user', 'password'; commit;</code>
9	Grant privileges: <code>grant dba, resource to <user>; commit;</code>

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Learn More:

- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

3.3.2 Set Up an ODBC Driver for a Connection to a Repository

Setting up an ODBC driver is only necessary for certain database management systems, which require specific set-up in order to operate as a repository.

Other database management systems connect through OLE DB and do not require a driver. However, you can connect to an Oracle repository using either OLE DB or an ODBC driver; if you intend to connect through the ODBC driver, you must first set it up.

For more information and the steps involved for specific systems, follow the links below:

Topic	Link
MySQL	Set up a MySQL ODBC Driver ^[154]
PostgreSQL	Set up a PostgreSQL ODBC Driver ^[155]
Adaptive Server Anywhere	Set up an Adaptive Server Anywhere ODBC Driver ^[158]
Progress OpenEdge	Set up a Progress OpenEdge ODBC Driver ^[159]
Oracle	Set up an Oracle ODBC Driver ^[160]

Notes:

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: \Windows\SysWOW64\odbcad32.exe

3.3.2.1 MySQL ODBC Driver

This topic explains how to set up a MySQL ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- MySQL DBMS and repository
- MySQL ODBC driver software version 5.1.5 (version 3.51.14 creates problems in incorporating tests in elements)

Use to:

- Set up the MySQL ODBC driver so that you can connect to your MySQL repository in Enterprise Architect

How to:

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option The ODBC Data Source Administrator window displays
2	Click on the Add button The Create New Data Source dialog displays, enabling you to add a new DSN
3	Select MySQL ODBC 5.1 Driver from the list
4	Click on the Finish button The MySQL Connector/ODBC dialog displays
5	Enter the following configuration details: <ul style="list-style-type: none"> • A data source name for the connection • A description (optional) • The host address of the DBMS server • User name and password • The database name on the selected server
6	To set the advanced options, click on the Details>> button
7	Select the following checkboxes (where provided): <ul style="list-style-type: none"> • Return matched rows instead of affected rows (Connection or Cursors/Results tab) • Allow big result sets (Connection tab)
8	Click on the Test Connection button to confirm that the details are correct
9	<ul style="list-style-type: none"> • If the test succeeds, click on the OK button to complete the configuration • If the test does not succeed, review your settings

Your MySQL connection is now available to use in Enterprise Architect.

Notes:

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: `Windows\SysWOW64\odbcad32.exe`

Learn More:

- [Create MySQL Repository](#) ^[149]
- [Connect To MySQL Data Repository](#) ^[162]

3.3.2.2 PostgreSQL ODBC Driver

This topic explains how to set up a PostgreSQL ODBC driver.

Prerequisites:

You must first have installed:

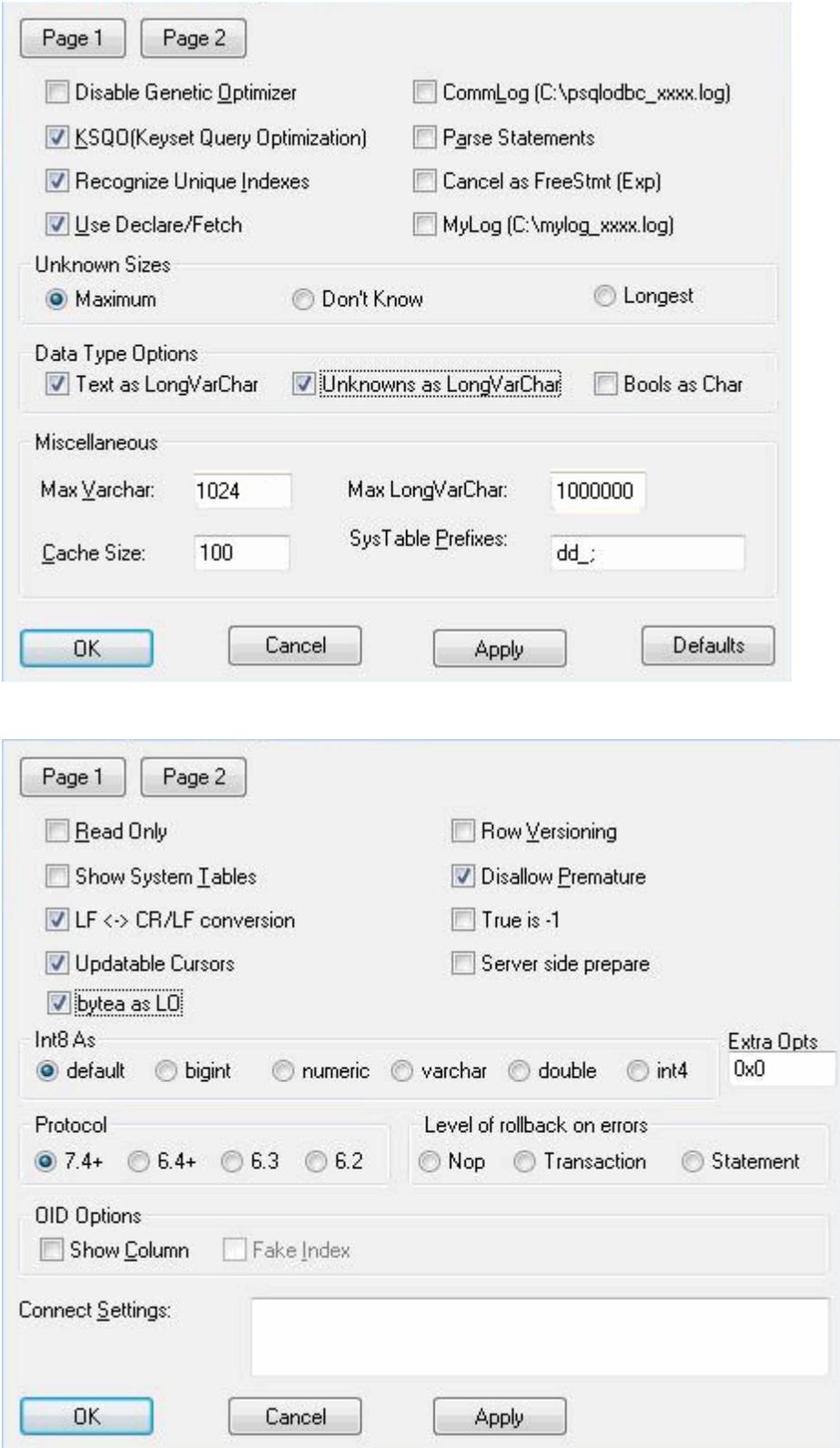
- Microsoft MDAC components
- PostgreSQL DBMS and repository
- PostgreSQL ODBC driver software version 7.03.01.00 or above (note that versions 8.3.4 and 8.4.1 of the PostgreSQL ODBC Driver are not supported)

Use to:

- Set up the PostgreSQL ODBC driver so that you can connect to your PostgreSQL repository in Enterprise Architect

How to:

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option The ODBC Data Source Administrator window displays
2	Click on the Add button The Create New Data Source dialog displays, enabling you to add a new DSN
3	Select PostgreSQL UNICODE from the list
4	Click on the Finish button The Postgre SQL Connector/ODBC dialog displays
5	Enter the following configuration details: <ul style="list-style-type: none"> • A data source name for the connection • The actual name of the database • The host address of the DBMS server • User name • A description (optional) • The password
6	To set the advanced options, click on the Datasource button and set the options on Page 1 and Page 2 as shown below:

Step	Action
	
7	<p>If you are using PostgreSQL version 8+, on Page 2 select:</p> <ul style="list-style-type: none"> The Disallow Premature checkbox

Step	Action
	<ul style="list-style-type: none"> In the Protocol panel, the 7.4+ radio button
8	Click on the OK button to complete the configuration

Your PostgreSQL connection is now available to use in Enterprise Architect.

Notes:

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: *Windows\SysWOW64\odbcad32.exe*

Learn More:

- [Create a PostgreSQL Repository](#)^[15]
- [Connect To a PostgreSQL Data Repository](#)^[16]

3.3.2.3 ASA ODBC Driver

This topic explains how to set up an Adaptive Server Anywhere (ASA) ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- Adaptive Server Anywhere - SQL Anywhere Studio 8 or higher, and created a repository
- ASA ODBC driver software (installed with the ASA DBMS)

Use to:

- Set up the ASA ODBC driver so that you can connect to your ASA repository in Enterprise Architect

How to:

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option The ODBC Data Source Administrator window displays
2	Click on the Add button The Create New Data Source dialog displays, enabling you to add a new DSN
3	Select Adaptive Server Anywhere or SQL Anywhere from the list
4	Click on the Finish button The ASA Connector/ODBC dialog displays
5	Enter the following configuration details: <ul style="list-style-type: none"> A data source name for the connection, on the ODBC tab User name and password on the Login tab (dba, sql are the defaults when ASA is installed) The server name and the path to the database, on the Database tab

Step	Action
	<ul style="list-style-type: none"> The network protocol on the Network tab (if the database is on a network location)
6	Return to the ODBC tab and test the connection
7	Click on the OK button to complete the configuration

Your ASA connection is now available to use in Enterprise Architect.

Notes:

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: `Windows\SysWOW64\odbcad32.exe`

Learn More:

- [Create an Adaptive Server Anywhere Repository^{\[152\]}](#)
- [Connect To an ASA Data Repository^{\[169\]}](#)

3.3.2.4 Progress OpenEdge ODBC Driver

This topic explains how to set up a Progress OpenEdge ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- OpenEdge DBMS and repository
- DataDirect ODBC driver for OpenEdge software (version 4.20 minimum)

Use to:

- Set up the Progress OpenEdge ODBC driver so that you can connect to your OpenEdge repository in Enterprise Architect

How to:

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option The ODBC Data Source Administrator window displays
2	Click on the Add button The Create New Data Source dialog displays, enabling you to add a new DSN
3	Select DataDirect/OpenEdge SQL Driver from the list
4	Click on the Finish button The DSN Configuration dialog displays
5	Enter the following configuration details: <ul style="list-style-type: none"> A data source name for the connection

Step	Action
	<ul style="list-style-type: none"> • A description (optional) • The host name and port number of the DBMS server • The database name on the selected server • The User ID
6	Click on the Test Connect button to confirm that the details are correct
7	<ul style="list-style-type: none"> • If the test succeeds, click on the OK button to complete the configuration • If the test does not succeed, review your settings

Your OpenEdge connection is now available to use in Enterprise Architect.

Notes:

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: `Windows\SysWOW64\odbcad32.exe`

Learn More:

- [Create a PostgreSQL Repository](#)^[15]
- [Connect To a PostgreSQL Data Repository](#)^[16]

3.3.2.5 Oracle ODBC Driver

This topic explains how to set up an Oracle ODBC driver.

Prerequisites:

You must first have installed:

- Microsoft MDAC components
- Oracle DBMS and repository

Use to:

- Set up the Oracle ODBC driver so that you can connect to your Oracle repository in Enterprise Architect, in preference to using an OLE DB connection

How to:

Step	Action
1	Select the Windows™ Control Panel Administrative Tools Data Sources (ODBC) option The ODBC Data Source Administrator window displays
2	Click on the Add button The Create New Data Source dialog displays, enabling you to add a new DSN
3	Select Oracle in OraDB11g_home1 from the list (or similar, depending on the ODBC installation)
4	Click on the Finish button The Oracle ODBC Driver Configuration dialog displays

Step	Action
5	Enter the following configuration details: <ul style="list-style-type: none"> • A data source name for the connection • A description (optional) • The TNS Service Name (click on the drop down arrow and select from the list) • User ID
6	Click on the Test Connection button and enter the Oracle user password to confirm that the details are correct
7	<ul style="list-style-type: none"> • If the test succeeds, click on the OK button to complete the configuration • If the test does not succeed, review your settings

Your Oracle connection is now available to use in Enterprise Architect.

Notes:

- Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on 64-bit clients, run the 32-bit ODBC Data Source Administrator from C: `Windows\SysWOW64\odbcad32.exe`

Learn More:

- [Create an Oracle Server Repository](#)^[151]
- [Connect to an Oracle Data Repository \(ODBC\)](#)^[164]

3.3.3 Connect to a Data Repository

This topic describes how to connect to a data repository to:

- Access an Enterprise Architect model in the repository
- Reverse engineer a database schema into a model using the ODBC driver provided by the DBMS vendor

To connect to a repository, you must have the usual SELECT, UPDATE, INSERT and DELETE permissions.

For more information and the steps involved for specific systems, follow the links below:

Topic	Link
MySQL	Connect to a MySQL Data Repository ^[162]
SQL Server	Connect To a SQL Server Data Repository ^[163]
Oracle	Connect To an Oracle Data Repository ^[164]
PostgreSQL	Connect To a PostgreSQL Data Repository ^[167]
Adaptive Server Anywhere	Connect To an Adaptive Server Anywhere Data Repository ^[169]
MSDE Server	Connect To an MSDE Server data Repository ^[170]
Progress OpenEdge	Connect To a Progress OpenEdge Data Repository ^[170]

Learn More:

- [Import Database Schema Using ODBC](#)^[1375]

3.3.3.1 MySQL Data Repository

This topic explains how to connect to a MySQL data repository in Enterprise Architect.

Access: **File | Open Project (Ctrl+O)**

Use to:

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create a MySQL Repository
- Set up a MySQL ODBC driver

How to:

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox
2	Click on the (...) (Browse) button The Data Link Properties dialog displays
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list
4	Click on the Next>> button The Connection tab displays
5	Click on the Use data source name radio button and on the drop-down arrow in its field From the list, select the ODBC driver you have set up to connect to your MySQL repository
6	If required, type in a User name and Password
7	If required, type in an initial catalog
8	Click on the Test Connection button to confirm that the details are correct
9	If the test does not succeed, revise your settings If the test succeeds, click on the OK button; the Connection Name & Type dialog displays
10	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog
11	If required, select the Encrypt Connection String checkbox This encrypts and hides the connection details of the database from the users that the connection string is given to
12	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model
13	If required, select the Use WAN Optimization checkbox

Step	Action
	To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 17
14	In the Server field, type the network name or address of the optimizer server
15	In the Port field, type the port on which the server is running on the remote machine
16	In the DSN field, type the data source name of the database as it appears on the remote machine.
17	Click on the OK button to complete the configuration

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Create a MySQL Repository](#)^[149]
- [Set up a MySQL ODBC driver](#)^[154]
- [The WAN Optimizer](#)^[183]
- [Open a Project](#)^[138]

3.3.3.2 SQL Server Data Repository

This topic explains how to connect to a SQL Server data repository in Enterprise Architect.

Access: **File | Open Project (Ctrl+O)**

Use to:

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create a SQL Server Repository

How to:

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox
2	Click on the (...) (Browse) button The Data Link Properties dialog displays
3	Select Microsoft OLE DB Provider for SQL Server from the list
4	Click on the Next>> button The Connection tab displays

Step	Action
5	Type in the server details, including Server Name, User Name and Password.
6	Click on the Select the database on the server option and on the drop-down arrow. From the list, select the model to connect to.
7	Click on the Test Connection button to confirm that the details are correct
8	If the test does not succeed, revise your settings If the test succeeds, click on the OK button; the Connection Name & Type dialog displays
9	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog
10	If required, select the Encrypt Connection String checkbox This encrypts and hides the connection details of the database from the users that the connection string is given to
11	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model
12	If required, select the Use WAN Optimization checkbox To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time If you select this checkbox, complete the next two fields (see your administrator for the correct values); otherwise go to step 15
13	In the Server field, type the network name or address of the optimizer server
14	In the Port field, type the port on which the server is running on the remote machine
15	Click on the OK button to complete the configuration

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Create a SQL Server Repository](#)^[150]
- [The WAN Optimizer](#)^[183]
- [Open a Project](#)^[138]

3.3.3.3 Oracle Data Repository (ODBC)

This topic explains how to connect to an Oracle 9i, 10g or 11g data repository in Enterprise Architect using an Oracle ODBC driver.

You can also connect to the Oracle repository using OLE DB.

Access: **File | Open Project (Ctrl+O)**

Use to:

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create an Oracle Repository
- Set up an Oracle ODBC driver

How to:

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox
2	Click on the (...) (Browse) button The Data Link Properties dialog displays
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list
4	Click on the Next>> button The Connection tab displays
5	In the Data source field, click on the drop-down arrow and select the data source name
6	Type in the User name and Password
7	Click on the Test Connection button to confirm that the details are correct
8	If the test does not succeed, revise your settings If the test succeeds, click on the OK button; Oracle prompts you for the password
9	Type in the password The Connection Name and Type dialog displays
10	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog
11	If required, select the Encrypt Connection String checkbox This encrypts and hides the connection details of the database from the users that the connection string is given to
12	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model
13	If required, select the Use WAN Optimization checkbox To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time If you select this checkbox, complete the next two fields (see your administrator for the correct values); otherwise go to step 16
14	In the Server field, type the network name or address of the optimizer server
15	In the Port field, type the port on which the server is running on the remote machine

Step	Action
16	Click on the OK button to complete the configuration

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Create an Oracle Data Repository](#)^[151]
- [Set up an Oracle ODBC Driver](#)^[160]
- [Connect to an Oracle Data Repository \(OLE DB\)](#)^[166]
- [The WAN Optimizer](#)^[183]
- [Open a Project](#)^[138]

3.3.3.4 Oracle Data Repository (OLE DB)

This topic explains how to connect to an Oracle 9i, 10g or 11g data repository in Enterprise Architect using OLE DB.

You can also connect to the Oracle repository using an Oracle ODBC driver.

Access: **File | Open Project (Ctrl+O)**

Use to:

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create an Oracle Repository

How to:

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox
2	Click on the (...) (Browse) button The Data Link Properties dialog displays
3	Select Oracle Provider for OLE DB from the list Do not select Microsoft OLE DB Provider for Oracle ; Enterprise Architect might not work as expected
4	Click on the Next>> button The Connection tab displays
5	In the Data source field, click on the drop-down arrow and select the data source name (the service name of the Oracle database)

Step	Action
6	Type in the User name and Password
7	Click on the Test Connection button to confirm that the details are correct
8	If the test does not succeed, revise your settings If the test succeeds, click on the OK button; the Connection Name and Type dialog displays
9	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog
10	If required, select the Encrypt Connection String checkbox This encrypts and hides the connection details of the database from the users that the connection string is given to
11	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model
12	If required, select the Use WAN Optimization checkbox To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time If you select this checkbox, complete the next two fields (see your administrator for the correct values); otherwise go to step 15
13	In the Server field, type the network name or address of the optimizer server
14	In the Port field, type the port on which the server is running on the remote machine
15	Click on the OK button to complete the configuration

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Create an Oracle Data Repository](#)^[151]
- [Connect to an Oracle Data Repository \(ODBC\)](#)^[164]
- [The WAN Optimizer](#)^[183]
- [Open a Project](#)^[138]

3.3.3.5 PostgreSQL Data Repository

This topic explains how to connect to a PostgreSQL data repository in Enterprise Architect.

Access: **File | Open Project (Ctrl+O)**

Use to:

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create a PostgreSQL Repository
- Set up a PostgreSQL ODBC driver

How to:

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox
2	Click on the (...) (Browse) button The Data Link Properties dialog displays
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list
4	Click on the Next>> button The Connection tab displays
5	Click on the Use data source name radio button and on the drop-down arrow in its field From the list, select the ODBC driver you have set up to connect to your PostgreSQL repository
6	Click on the Test Connection button to confirm that the details are correct
7	If the test does not succeed, revise your settings If the test succeeds, click on the OK button; the Connection Name & Type dialog displays
8	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog
9	If required, select the Encrypt Connection String checkbox This encrypts and hides the connection details of the database from the users that the connection string is given to
10	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model
11	If required, select the Use WAN Optimization checkbox To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 15
12	In the Server field, type the network name or address of the optimizer server
13	In the Port field, type the port on which the server is running on the remote machine
14	In the DSN field, type the data source name of the database as it appears on the remote machine.
15	Click on the OK button to complete the configuration

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Create a PostgreSQL Repository](#)^[151]
- [Set up a PostgreSQL ODBC Driver](#)^[155]
- [The WAN Optimizer](#)^[183]
- [Open a Project](#)^[138]

3.3.3.6 ASA Data Repository

This topic explains how to connect to an Adaptive Server Anywhere data repository in Enterprise Architect.

Access: **File | Open Project (Ctrl+O)**

Use to:

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create an ASA Data Repository
- Set up an ASA ODBC driver

How to:

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox
2	Click on the (...) (Browse) button The Data Link Properties dialog displays
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list
4	Click on the Next>> button The Connection tab displays
5	Click on the Use data source name radio button and on the drop-down arrow in its field From the list, select the ODBC driver you have set up to connect to your ASA repository
6	Click on the Test Connection button to confirm that the details are correct
7	If the test does not succeed, revise your settings If the test succeeds, click on the OK button; the Connection Name & Type dialog displays
8	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog
9	If required, select the Encrypt Connection String checkbox This encrypts and hides the connection details of the database from the users that the connection string is given to
10	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded;

Step	Action
	instead, only the parts that are necessary to display the visible portion of the tree are loaded With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model
11	If required, select the Use WAN Optimization checkbox To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 15
12	In the Server field, type the network name or address of the optimizer server
13	In the Port field, type the port on which the server is running on the remote machine
14	In the DSN field, type the data source name of the database as it appears on the remote machine.
15	Click on the OK button to complete the configuration

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Create an Adaptive Server Anywhere Repository](#)^[152]
- [Set up an ASA ODBC Driver](#)^[158]
- [The WAN Optimizer](#)^[183]
- [Open a Project](#)^[138]

3.3.3.7 MSDE Server Data Repository

To connect to an MSDE Server Data Repository, follow the steps in [Connect to an SQL Server Repository](#)^[163].

Use to:

- Connect to the data repository so that you can open a project held in the repository

3.3.3.8 Progress OpenEdge Repository

This topic explains how to connect to a Progress OpenEdge data repository in Enterprise Architect.

Access: **File | Open Project (Ctrl+O)**

Use to:

- Connect to the data repository so that you can open a project held in the repository

Prerequisites:

- Create a Progress OpenEdge Repository
- Set up a Progress OpenEdge ODBC driver

How to:

Step	Action
1	In the Open Project dialog, select the Connect to Server checkbox
2	Click on the (...) (Browse) button The Data Link Properties dialog displays
3	Select Microsoft OLE DB Provider for ODBC Drivers from the list
4	Click on the Next>> button The Connection tab displays
5	Click on the Use data source name radio button and on the drop-down arrow in its field From the list, select the ODBC driver you have set up to connect to your Progress OpenEdge repository
6	Enter the User Name and Password
7	Enter the Initial catalog
8	Click on the All tab and double-click on Extended Properties
9	Click on the Connection tab again
10	In the Property Value field, edit the value to: DefaultSchema=PUB Click on the Test Connection button to confirm that the details are correct
11	If the test does not succeed, revise your settings If the test succeeds, click on the OK button; the Logon to Progress dialog displays
12	Check the details to ensure that they are correct Click on the OK button; the Connection Name and Type dialog displays
13	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog
14	If required, select the Encrypt Connection String checkbox This encrypts and hides the connection details of the database from the users that the connection string is given to
15	If required, select the Lazy Load checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model
16	If required, select the Use WAN Optimization checkbox To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 20
17	In the Server field, type the network name or address of the optimizer server
18	In the Port field, type the port on which the server is running on the remote machine

Step	Action
19	In the DSN field, type the data source name of the database as it appears on the remote machine.
20	Click on the OK button to complete the configuration

Notes:

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Create a Progress OpenEdge Repository](#)^[153]
- [Set up a Progress OpenEdge ODBC Driver](#)^[159]
- [The WAN Optimizer](#)^[183]
- [Open a Project](#)^[138]

3.3.4 Upsizing

This topic explains how you upsize your .EAP project or template to one of a range of DBMS repositories.

For more information and the steps involved for specific systems, follow the links below:

Topic	Link
MySQL	Upsize to MySQL ^[172]
Access	Upsize to Access 2007 ^[174]
SQL Server	Upsize to SQL Server ^[176]
Oracle	Upsize to Oracle 9i, 10g or 11g ^[177]
PostgreSQL	Upsize to PostgreSQL ^[179]
Adaptive Server Anywhere	Upsize to Sybase ASA ^[180]
MSDE Server	Upsize to MSDE ^[182]
Progress OpenEdge	Upsize to Progress OpenEdge ^[182]

Notes:

- You cannot move a model from a source .EAP file of a version earlier than 3.5.0
- Before proceeding, ensure MDAC 2.6 or higher is installed on your system

3.3.4.1 Upsize to MySQL

This topic explains how to upsize your .EAP project or template for use with MySQL.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Install MySQL components
- Stage 3: Select table type
- Stage 4: Create an empty database repository
- Stage 5: Set up the ODBC Driver

- Stage 6: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file	
2	Select and run Tools Data Management Project Integrity Check This ensures your project data is 'clean' before uploading	Project Integrity Check ^[414]

Install MySQL Components:

Step	Action	See also
1	Install MySQL version 4.0.3 or higher	Create a MySQL Repository ^[149]
2	Install MySQL ODBC 3.51 or higher	

Select Table Type:

If MyISAM table types are used (default), transactional support is disabled. To enable transactions you must set up MySQL to use InnoDB tables and create the database tables as InnoDB type.

Sparx provide scripts to create both InnoDB- and MyISAM-based repository tables.

Step	Action	See also
1	Set up the <i>MySQL .ini</i> file as required	
2	If you are using <i>InnoDB</i> tables run the <i>MySQL_InnoDB_EASchema.sql</i> script If you are using <i>MyISAM</i> tables run the <i>MySQL_MyISAM_EASchema.sql</i> script.	Sparx Systems website

Create Database Repository:

Step	Action	See also
1	Create an empty database repository	Create a New MySQL Repository ^[149]

Set Up the ODBC Driver:

Step	Action	See also
1	Create a suitable ODBC Data Source to point to your new database Select the following checkboxes: <ul style="list-style-type: none"> • Return matched rows instead of affected rows • Allow big result sets 	Set up a MySQL ODBC Driver ^[154]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect Click on the Cancel button on the Open Project screen to open with no project loaded	
2	Select the Tools Data Management Project Transfer menu option The Project Transfer dialog displays	
3	In the Transfer Type panel, select .EAP to DBMS	
4	In the Source Project field, type the name of the .EAP file to upsize to MySQL	
5	At the right of the Target Project field, click on the (...) (Browse) button The Datalink Properties dialog displays	
6	Select Microsoft OLE DB Provider for ODBC Drivers from the list Click on the Next button	
7	In the Use Data source name field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database Click on the OK button	Connect to a MySQL Data Repository ^[162]
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file	
9	Click on the Transfer button to begin the data transfer process	

When the process is complete, you have upsized your model to MySQL and can now open it from Enterprise Architect.

3.3.4.2 Upsize to Access 2007

This topic explains how to upsize your .EAP project or template for use with Access 2007.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file	
2	Select and run Tools Data Management Project Integrity Check This ensures your project data is 'clean' before uploading	Project Integrity Check ^[414]

Create Database Repository:

Step	Action	See also
1	Create an empty database repository	Create an Access 2007 Repository ^[149]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect Click on the Cancel button on the Open Project screen to open with no project loaded	
2	Select the Tools Data Management Project Transfer menu option The Project Transfer dialog displays	
3	In the Transfer Type panel, select .EAP to DBMS	
4	In the Source Project field, type the name of the .EAP file to upsize to Access 2007	
5	At the right of the Target Project field, click on the (...) (Browse) button The Datalink Properties dialog displays	
6	Select Microsoft Office 12.0 Access Database Engine OLE DB Provider from the list Click on the Next button	
7	On the Data Source Details page of the Connection dialog, type in the full path to the Access 2007 .ACCDB file Click on the OK button to return to the Project Transfer dialog	
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file	
9	Click on the Transfer button to begin the data transfer process	

When the process is complete, you have upsized your model to Access 2007 and can now open it from Enterprise Architect.

3.3.4.3 Upsize to SQL Server

This topic explains how to upsize your .EAP project or template for use with SQL Server.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file	
2	Select and run Tools Data Management Project Integrity Check This ensures your project data is 'clean' before uploading	Project Integrity Check ^[414]

Create Database Repository:

Step	Action	See also
1	Install SQL Server	
2	Create the empty database	Create an SQL Server Repository ^[150]
3	Using a tool such as the SQL Query Analyser, load the <i>SQLServer_EASchema.sql</i> file	Sparx Systems Website
4	Make sure the new database is the currently active database	
5	Run the script to create all required data structures	Create an SQL Server Repository ^[150]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect Click on the Cancel button on the Open Project screen to open with no project loaded	
2	Select the Tools Data Management Project Transfer menu option The Project Transfer dialog displays	
3	In the Transfer Type panel, select .EAP to DBMS	

Step	Action	See also
4	In the Source Project field, type the name of the .EAP file to upsize to SQL Server	
5	At the right of the Target Project field, click on the (...) (Browse) button The Datalink Properties dialog displays	
6	Select Microsoft OLE DB Provider for SQL Server from the list Click on the Next button	
7	On the Data Source Details page of the Connection dialog, type in the server name, database name and security details as required Click on the OK button	Connect to SQL Server Data Repository ^[163]
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file	
9	Click on the Transfer button to begin the data transfer process	

When the process is complete, you have upsized your model to SQL Server and can now open it from Enterprise Architect.

Notes:

- When transferring a project you must have **db_ddladmin** permission in order to execute the **SET IDENTITY_INSERT (table) {ON | OFF}** command

3.3.4.4 Upsize to Oracle 9i, 10g or 11g

This topic explains how to upsize your .EAP project or template for use with Oracle 9i, 10g or 11g.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file	
2	Select and run Tools Data Management Project Integrity Check This ensures your project data is 'clean' before uploading	Project Integrity Check ^[414]

Create Database Repository:

Step	Action	See also
1	Install Oracle	
2	Create the empty database	Create an Oracle Data Repository ^[15]
3	Using a tool such as the SQL*Plus or SQL Plus Worksheet, load the <i>Oracle_EASchema.sql</i> file	Sparx Systems Website
4	Make sure the new database is the currently active database	
5	Run the script to create all required data structures	Create an Oracle Data Repository ^[15]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect Click on the Cancel button on the Open Project screen to open with no project loaded	
2	Select the Tools Data Management Project Transfer menu option The Project Transfer dialog displays	
3	In the Transfer Type panel, select .EAP to DBMS	
4	In the Source Project field, type the name of the .EAP file to upsize to Oracle	
5	At the right of the Target Project field, click on the (...) (Browse) button The Datalink Properties dialog displays	
6	Select Oracle Provider for OLE DB from the list Click on the Next button	
7	On the Connection page of the Datalink Properties dialog, enter the Oracle service name in the Data Source field, and the user name and password as required. Click on the OK button	Connect to an Oracle Data Repository ^[16]
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file	
9	Click on the Transfer button to begin the data transfer process	

When the process is complete, you have upsized your model to Oracle and can now open it from Enterprise Architect.

Notes:

- When transferring a project to Oracle you must have permission to execute the **CREATE SEQUENCE**

command

3.3.4.5 Upsize to PostgreSQL

This topic explains how to upsize your .EAP project or template for use with PostgreSQL.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Install PostgreSQL Driver
- Stage 4: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file	
2	Select and run Tools Data Management Project Integrity Check This ensures your project data is 'clean' before uploading	Project Integrity Check ^[414]

Create Database Repository:

Step	Action	See also
1	Install PostgreSQL, version 7.3.2 or higher	
2	Create the empty database	Create a PostgreSQL Repository ^[151]
3	From the PSQL command line, or using a tool such as the PostgreSQL command line, pgAdminIII or EMS PostgreSQL Manager, load the <i>PostgreSQL_EASchema.sql</i> file	Sparx Systems Website
4	Run the script to create all required data structures	Create a PostgreSQL Repository ^[151]

Set Up the ODBC Driver:

Step	Action	See also
1	Install psqLODBC, version 7.03.01.00 or higher (but do not use version 8.4.1)	
2	Create a suitable ODBC Data Source to point to your new database	Set up a PostgreSQL ODBC Driver ^[155]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect Click on the Cancel button on the Open Project screen to open with no project loaded	
2	Select the Tools Data Management Project Transfer menu option The Project Transfer dialog displays	
3	In the Transfer Type panel, select .EAP to DBMS	
4	In the Source Project field, type the name of the .EAP file to upsize to SQL Server	
5	At the right of the Target Project field, click on the (...) (Browse) button The Datalink Properties dialog displays	
6	Select Microsoft OLE DB Provider for ODBC Drivers from the list Click on the Next button	
7	On the Use Data Source Name field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database Click on the OK button	Connect to a PostgreSQL Data Repository ¹⁶⁷
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file	
9	Click on the Transfer button to begin the data transfer process	

When the process is complete, you have upsized your model to PostgreSQL and can now open it from Enterprise Architect.

Notes:

- During the transfer, if an error message displays reporting '*...nonstandard use of \ in a string literal...*', set the server variable in the *postgresql.conf* file to: *escape_string_warning = off*

3.3.4.6 Upsize to Sybase ASA

This topic explains how to upsize your .EAP project or template for use with Sybase Adaptive Server Anywhere.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file	
2	Select and run Tools Data Management Project Integrity Check This ensures your project data is 'clean' before uploading	Project Integrity Check ^[414]

Create Database Repository:

Step	Action	See also
1	Install Adaptive Server Anywhere - SQL Anywhere Studio 8 or higher This also installs the ASA ODBC driver	
2	Using Sybase Central, create a new database for the Enterprise Architect repository	Create an Adaptive Server Anywhere Repository ^[152]
3	Open Interactive SQL and load the <i>ASA_EASchema.sql</i> file	Sparx Systems Website
4	Run the script to create all required data structures	Create an Adaptive Server Anywhere Repository ^[152]
5	Create a suitable ODBC Data Source to point to your new database	Set up an Adaptive Server Anywhere ODBC Driver ^[158]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect Click on the Cancel button on the Open Project screen to open with no project loaded	
2	Select the Tools Data Management Project Transfer menu option The Project Transfer dialog displays	
3	In the Transfer Type panel, select .EAP to DBMS	
4	In the Source Project field, type the name of the .EAP file to upsize to ASA	
5	At the right of the Target Project field, click on the (...) (Browse) button The Datalink Properties dialog displays	
6	Select Microsoft OLE DB Provider for ODBC Drivers from the list Click on the Next button	
7	In the Use Data source name field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database	Connect to an Adaptive Server Anywhere Data Repository ^[169]

Step	Action	See also
	Click on the OK button	
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file	
9	Click on the Transfer button to begin the data transfer process	

When the process is complete, you have upsized your model to Adaptive Server Anywhere and can now open it from Enterprise Architect.

3.3.4.7 Upsize to MSDE

To upsize your model to MSDE, follow the steps in [Upsize to SQL Server](#)^[176].

3.3.4.8 Upsize to Progress OpenEdge

This topic explains how to upsize your .EAP project or template for use with Progress OpenEdge.

The process has the following stages:

- Stage 1: Perform project integrity check
- Stage 2: Create an empty database repository
- Stage 3: Transfer the data

Perform Project Integrity Check:

Step	Action	See also
1	Open the base project or template .eap file	
2	Select and run Tools Data Management Project Integrity Check This ensures your project data is 'clean' before uploading	Project Integrity Check ^[414]

Create Database Repository:

Step	Action	See also
1	Install OpenEdge, version 10.0B3 or higher	
2	Create an empty OpenEdge database for the new repository	Create a Progress OpenEdge Repository ^[153]
3	Download the <i>OpenEdge_EASchema.sql</i> file from the Sparx Systems Registered Users Resources page	Sparx Systems Website
4	Make sure the new database is selected as the current database	
5	Run the script to create all required data structures	Create a Progress OpenEdge Repository ^[153]
6	Install OpenEdge ODBC Driver, 10.0B or higher	Set Up a Progress OpenEdge

Step	Action	See also
	Create a suitable ODBC Data Source to point to your new database	ODBC Driver ^[159]

Transfer the Project Data:

Step	Action	See also
1	Open Enterprise Architect Click on the Cancel button on the Open Project screen to open with no project loaded	
2	Select the Tools Data Management Project Transfer menu option The Project Transfer dialog displays	
3	In the Transfer Type panel, select .EAP to DBMS	
4	In the Source Project field, type the name of the .EAP file to upsize to Progress OpenEdge	
5	At the right of the Target Project field, click on the (...) (Browse) button The Datalink Properties dialog displays	
6	Select Microsoft OLE DB Provider for ODBC Drivers from the list Click on the Next button	
7	In the Use Data source name field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database Click on the OK button	Connect to a Progress OpenEdge Data Repository ^[170]
8	If required, select the Logfile checkbox and type a path and filename for the data transfer log file	
9	Click on the Transfer button to begin the data transfer process	

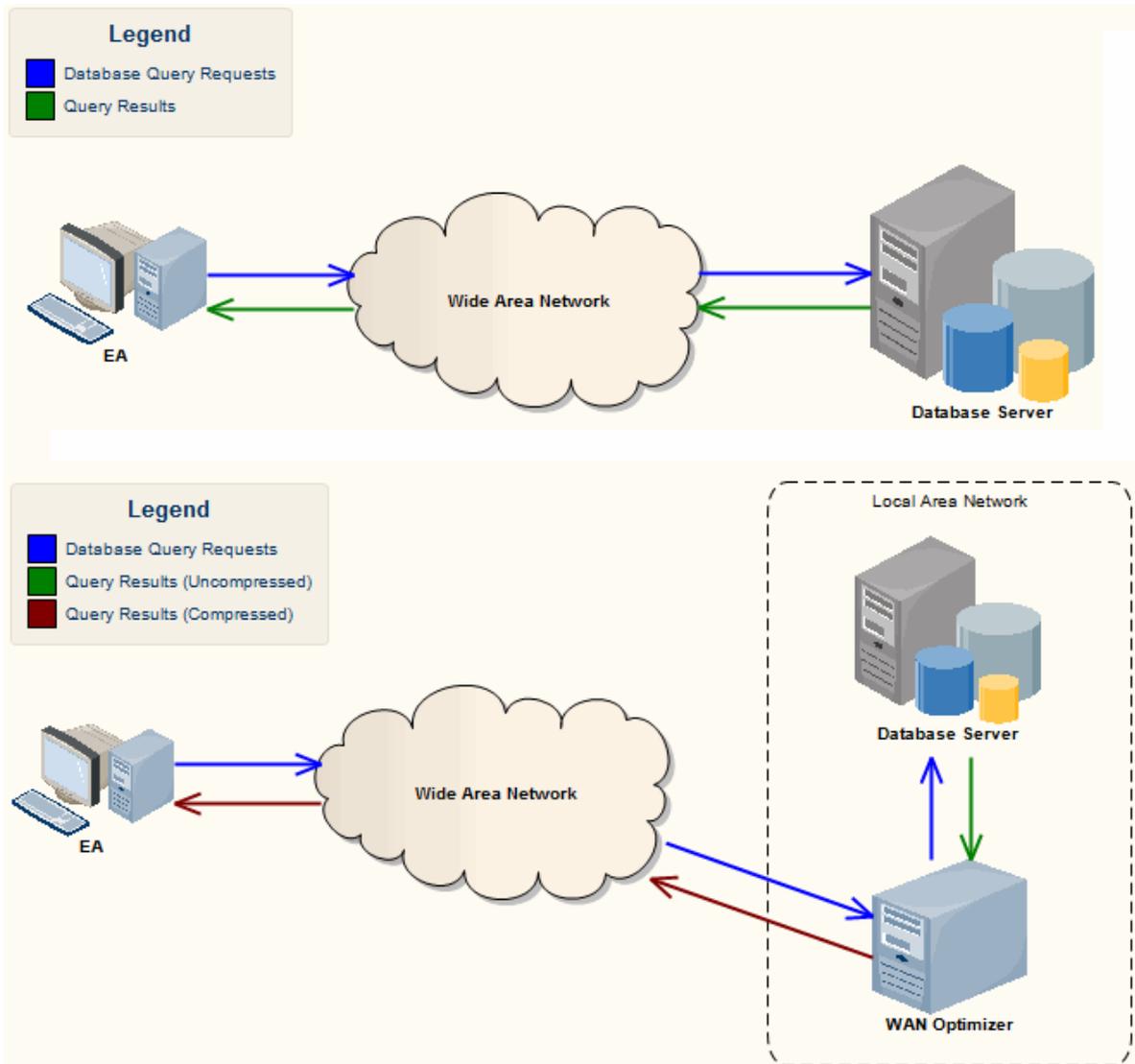
When the process is complete, you have upsized your model to Progress OpenEdge and can now open it from Enterprise Architect.

3.3.5 The WAN Optimizer

The Sparx *Wide Area Network (WAN) Optimizer* is a lightweight server installed on a Local Area Network (LAN) connection to a Database Management System (DBMS) that hosts an Enterprise Architect repository. You can configure the server to listen for client connections on a particular port; it acts as a local proxy to execute queries and return the results in a compressed format to the client.

The WAN Optimizer significantly improves Enterprise Architect's performance in a WAN by reducing the amount of data transmitted and, in turn, the number of network calls made.

In the following diagram, transmission between Enterprise Architect and a DBMS is depicted first without and then with the WAN Optimizer.



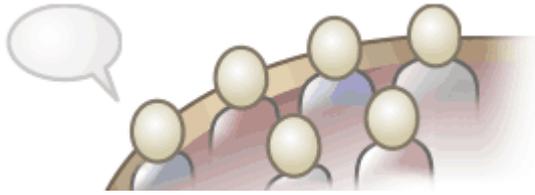
You can download the WAN Optimizer installer from the Downloads page of the Registered Users section of the Sparx Systems website. The Wan Optimizer Service installer package provides two installable features for the target machine:

- WAN Optimizer Service - the installer also helps register and start the service on the target machine, and add it to the Windows Startup folder
- WAN Optimizer Admin Client - to enable an administrator to administer and configure the service from a remote client.

The Optimizer has its own *Sparx WAN Optimizer User Guide*. See that User Guide for more information on:

- WAN Optimizer Components
- Installing and Starting the WAN Optimizer Service
- Configuring the Service
- Troubleshooting

3.4 Team Development



Enterprise Architect enables you to develop a project across a team of people, so that each person can access the latest data without the risk of damaging or losing that data.

Topics:

Topic	Detail	See also
Making project data available in a distributed environment	Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects in team-based and distributed development environments; for example: network deployment of model repositories, replication and XML Import/Export	Project Sharing ^[185]
Applying security to the model	Enterprise Architect Security is a means of improving collaborative design and development by preventing concurrent editing, and limiting the possibility of inadvertent model changes by users not designated as model authors	Configure User Security ^[194]
Using an internal discussion forum	The Enterprise Architect Project Team Review facility is a discussion forum within your development team community to discuss the development and progress of a project	Team Review Tools ^[216]
Building company policy and project development guidelines into the project	Enterprise Architect enables you to create workflow scripts that provide a robust approach to applying company policy and strengthening project development guidelines, by validating against the policy and procedures within the model itself	Workflow Scripts ^[231]
Standardizing and re-using project data	You can import and export Reference data (including Glossary and Issue information) from .XML files of another iteration of the same model, or of a different model	Sharing Reference Data ^[236]

3.4.1 Project Sharing

Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects in team-based and distributed development environments, through network deployment of model repositories, replication and XML Import/Export.

Topics:

Topic	Detail	See also
Network Deployment	Network deployment is possible under two different schemas: <ul style="list-style-type: none"> • .EAP based repositories or • DBMS server based repositories 	Share a Project ^[186]

Topic	Detail	See also
	<p>DBMS server based repositories offer better:</p> <ul style="list-style-type: none"> • response times than .EAP files on networks, due to the inherent structure of the DBMS • solutions when networking problems are encountered, as they have the ability to backtrack transactions caused by external breakdowns 	
Replication	<p>Replication enables data interchange between .EAP based repositories and is suitable for where many different users work independently in parallel development</p> <p>Modelers merge their changes into a Design Master only as required; it is recommended that a backup is carried out prior to replication</p> <p>Replication cannot be performed on repositories stored on a DBMS server</p>	Replication ^[188]
XMI Import and Export	<p>XMI Import/Export can be used to export and share discrete packages between developers; XMI enables the export of packages into XML files which can then be imported into any model</p> <p>Package control can be used to set up packages for version control and to enable batch export of packages using XMI; Version Control enables a repository to be maintained by a third-party source code control application that is used to control access and record revisions</p>	XMI Import/Export ^[319] Version Control ^[242] Controlled Packages ^[327]

Notes:

- DBMS Repository support is available with the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

Learn More:

- [Deployment of Enterprise Architect white paper](#)

3.4.1.1 Share Enterprise Architect Projects

The most efficient way of using Enterprise Architect to manage a team development is to *share* a project amongst a team of designers, developers and analysts.

Guide:

Topic	Detail	See also
By Sharing a project	<ul style="list-style-type: none"> • Many people can work on the model at the same time and contribute their particular skill • Team members can always see what the latest changes are, keeping the team informed and up to date with the project status 	Refresh View of Shared Project ^[187]
You can share a project in three	<ul style="list-style-type: none"> • Place the project in a shared network directory 	Share Projects on Network Drive ^[188]

Topic	Detail	See also
ways	<ul style="list-style-type: none"> • Use replication • Use a shared DBMS-based repository 	Replication ^[188] Server Based Repositories ^[148]

Notes:

- Project Sharing and Replication are enabled in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- DBMS repositories are supported in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

3.4.1.1 Refresh View of Shared Project

When a user of a shared model checks out a package and makes changes, other users can see those changes by refreshing their view of the package or the changed diagram within the package, in a number of ways.

Refresh	Detail	See also
Project	<ul style="list-style-type: none"> • Right-click on the package name in the Project Browser and select the Contents Reload Current Package context menu option • Select the File Reload Current Project menu option • Select the Reload Project icon in the Project toolbar • Press (Ctrl+Shift+F11) • Close the project and reopen it 	Contents Submenu ^[448] File ^[74] Project Toolbar ^[108]
Diagram	<ul style="list-style-type: none"> • Select the Window Reload Current View menu option • Right-click on the opened diagram tab in the diagram view, and select the Reload <diagram name> context menu option 	Window Menu ^[100] Diagram Tabs ^[543]

3.4.1.2 Share Projects on Network Drive

The easiest way to share a project amongst a work group of developers and analysts is to place the project file on a shared network drive, to which people connect from their workstations.

Individual developers and analysts can then open and work on the project concurrently.

Network Issues	See also
<p>Enterprise Architect accepts a number of concurrent connections without issue; however:</p> <ul style="list-style-type: none"> • There can be occasional 'lock-outs' when one user tries to access or update something another user is in the process of modifying • Changes to the Project Browser (and other project views) are not automatically updated; to compensate for this, users must occasionally reload their project to see changes made by other users • If two or more people work on the same diagram concurrently, unexpected results can occur; it is best to allow only one analyst to work on a diagram at a time 	Refresh Model View ^[187]

Network Issues	See also
<ul style="list-style-type: none"> (.EAP files only) If a user's machine crashes, the network suffers an outage or a machine is turned off unexpectedly, the project file might require repair to compensate for the sudden inconsistency; a repair facility is provided (Tools Data Management Manage .EAP File Repair .EAP File) to carry out this task 	Repair a Project ^[419]

3.4.1.3 Distributed Development

Enterprise Architect supports distributed development using two different techniques.

Topic	Detail	See also
Replication	Enables geographically separated analysts to update and modify parts of the model in replicas, then merge these back together at a central location	Replication ^[188]
XMI Import/Export	<p>Use XMI-based import/export to export discrete packages to XML to share among the development team; this has several benefits over replication:</p> <ul style="list-style-type: none"> You can assemble a model from only the parts necessary to get your job done You can assemble a full model if required You can assemble a model from different package versions for different purposes (such as customer visible, internal release only) You can roll-back parts of a model as required There is less chance of 'collisions' between developers if each works on a discrete package The process is controllable using a version control system, or through package control <p>XML based import/export is UML1.3 / XMI1.1 compliant; you can also write XML based tools to manipulate and extract information from XML files to enhance the development process</p> <p>XMI-based Import/Export is accessed through Project Model Import/Export</p>	Import/Export ^[319] Version Control ^[242] Controlled Packages ^[327] XML Import Export Options ^[817]

3.4.1.4 Replication

Apart from sharing Enterprise Architect projects in real time over a network, you can also share projects using replication.

Guide:

Topic	Detail	See also
Replication	Replication is a powerful means of sharing projects between isolated or mobile users; it enables different users to work independently of one another, and to merge their	

Topic	Detail	See also
	<p>changes at a later time</p> <p>In Replication:</p> <ul style="list-style-type: none"> • A project is converted to a design master, then replicas are made of the master • Users take the replicas away, modify the project, then bring their replicas back to be synchronized with the master file 	
Use Replication	<p>To use replication, follow the steps below</p> <ol style="list-style-type: none"> 1. Convert the base project to a design master 2. Create replicas from the design master 3. Take the replica away and work on it as required, then bring it back for synchronization with the design master 4. Synchronize the replicas, during which all changes to both the master and the replica are propagated in both directions, so they both finally contain the same information 	<p>Design Master ^[190]</p> <p>Create Replicas ^[190]</p> <p>Synchronize Replicas ^[191]</p>
Enterprise Architect Merge Rules	<p>Enterprise Architect follows these rules in merging:</p> <ul style="list-style-type: none"> • Additions are cumulative; that is, two replicas each creating three new Classes result in six new Classes after merging • Deletions prevail over modifications; if one replica changes a Class name and other deletes the Class, merging replicas results in both files losing the Class <p>Conflicting modifications appear in the Resolve Replication Conflicts dialog</p>	<p>Resolve Conflicts ^[193]</p>
Upgrades and Replicas	<p>When you upgrade your version of Enterprise Architect, you must not open a replica until you have opened the design master and then synchronized the replicas with the master; you cannot directly upgrade a replica</p>	<p>Upgrade Replicas ^[193]</p>
Avoid Change Collisions	<p>If two or more people make changes to the same element, Enterprise Architect arbitrarily overwrites one person's change with the other's; to avoid this, different users should work on different packages</p> <p>However, since Enterprise Architect does not enforce this rule, it is possible for users' work to conflict; to minimize difficulties, please note the following guidelines:</p> <ul style="list-style-type: none"> • If users are likely to have worked in the same area of the model, they should both witness the synchronization and confirm that they are happy with the net result • If small pieces of information have been lost, they should be typed into one of the merged models after synchronization • If a large piece of information has been lost (for example, an overwritten large Class note) use the Resolve Replication Conflicts dialog 	<p>Resolve Conflicts ^[193]</p>
Disable or	<p>If you have converted a project to a design master but now</p>	<p>Remove Replication ^[192]</p>

Topic	Detail	See also
Remove Replication Features	want to disable the replication features, you can remove Replication; ensure that you back up all your files first	

3.4.1.4.1 Design Masters

A design master is the first converted Enterprise Architect project that supports replication.

Access: [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Make Design Master](#)

Use to:

- Create the master project from which you create replicas that can be modified independently of the master project and re-merged later

How to:

To create a design master, follow the steps below:

Step	Action
1	Take a back-up of the required Enterprise Architect project
2	Select the project in the Project Browser
3	Select the Make Design Master menu option and follow the on-screen instructions

Learn More:

- [Create Replicas](#) ^[190]

3.4.1.4.2 Create Replicas

A replica is one of several copies of the design master of an Enterprise Architect project.

Access: [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Create New Replica](#)

Use to:

- Create a copy of the master project for you or another user to modify independently and re-merge later

How to:

To create a replica, follow the steps below:

Step	Action
1	First create a design master, then select the Create New Replica menu option and follow the on-screen instructions
2	Edit the replica over time and, when required, return the file for merging with the design master

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Replicas** permission to create a replica

Learn More:

- [Design Masters](#)^[190]
- [Manage Replicas User Permissions](#)^[205]

3.4.1.4.3 Synchronize Replicas

Synchronizing replicas combines the changes made to each file.

Access: [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Synchronize Replicas](#)

Use to:

- Merge the changes made to each replica with the design master, so that a new set of replicas with all changes can be generated and distributed
- Combine the changes made to two replicas, should it be necessary for two team members to combine their work

How to:

To synchronize replicas, follow the steps below:

Step	Action
1	Open the design master project file (or the first required replica)
2	Select the Synchronize Replicas menu option
3	Locate and select the (second) required replica to merge the open project and the replica

Notes:

- Information is copied both ways, including deletes, updates and inserts; both projects end up containing identical information
- If this process generates 'conflicting changes' errors, you should review and, if necessary, resolve these conflicts
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Replicas** permission to create a replica

Learn More:

- [Resolve Conflicts](#)^[193]
- [Manage Replicas User Permissions](#)^[205]

3.4.1.4.4 Remove Replication

Replication makes many changes to the database structure of your model, so the model file becomes considerably larger with additional information; you might, therefore, decide not to use the replication feature any more.

Access: [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Remove Replication](#)

Use to:

- Remove replication from your model

How to:

To remove replication from your model, follow the steps below:

Step	Action
1	Ensure that you have open a temporary repository (not the one having replication removed) - the menu option is not available if no repository is open
2	Select the Remove Replication menu option The Remove Replication Wizard displays
3	Enter the full path and file name of the project to have replication removed Click on the Next button
4	Enter the full path and file name of the base Enterprise Architect model (with no replication) to act as template Click on the Next button
5	Enter the full path and required file name for the output file Click on the Next button
6	Select whether to have a log file created and, if so, enter a file name for the log file
7	Click on the Run button to begin removing replication Enterprise Architect creates a new project containing all the model information Your model has now had replication removed, and should be considerably smaller

Notes:

- You cannot remove replication from a model with Auditing enabled - if you want to remove replication:
 - Disable Auditing
 - If prompted to do so, allow Enterprise Architect to roll back the database version
 - Remove replication

Learn More:

- [Auditing](#) ^[298]
- [Auditing Settings](#) ^[300]

3.4.1.4.5 Upgrade Replicas

A new release of Enterprise Architect might contain changes to the underlying project structure, such as more tables or changed queries. If you use Replication, you must take care with your upgrade.

Guide:

Topic	Detail	See also
Open Design Master Project first	After installing the new version of Enterprise Architect, it is very important that you open the <i>design master</i> before opening any of the replicas with the updated version	Design Masters ^[190]
Changes to the database design	Changes to the database design in a replicated project can ONLY be done to the design master; trying to update a replica at best does nothing, and at worst causes the update of the master to fail	
Propagate Changes	Design changes are propagated through to the replicas the next time the replicas are synchronized with the master	Synchronize Replicas ^[191]
Alternatively	One other strategy is to remove replication from a copy of the replica set, upgrade that project and convert it into a new design master from which new replicas are created	Remove Replication ^[192]

3.4.1.4.6 Resolve Conflicts

If two or more people each work on the same model object in their respective replicas between synchronizations, the replication engine has problems in resolving which change is the master.

Guide:

Topic	Detail	See also
Avoid the problem	Ensure that each team member always works in a separate area of the model within their replica	
Check for conflicts	After synchronizing replicas, open the Resolve Conflicts dialog (see below this table) and check if there were any conflicts	
Response to conflicts	<p>When a project record has been modified in different ways by different users, the replication engine selects one of the conflicting values based on rules within the JET replication manager</p> <p>However, the replication engine stores the discarded changes and flags the conflict on the Resolve Conflicts dialog so that you can choose to roll in the discarded change instead</p> <p>Normally it is not necessary or desirable to examine conflicts, since they represent relatively inconsequential pieces of information that can very easily be modified through the normal Enterprise Architect interface; for example, by moving a diagram element</p> <p>The only case in which the Resolve Conflicts dialog should be used is where a substantial piece of information has been overridden by a user, and you want to retrieve it</p>	

Access: [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Resolve Replication Conflicts](#)

Use to:

- Select which of two conflicting changes you should save to the Design Master and/or replicas, where a substantial piece of information has been overridden by a user and you want to retrieve it

How to:

To resolve conflicts in changes to the same model object in two separate replicas, follow the steps below:

Step	Action
1	Synchronize a replica with the Design Master, and display the Resolve Conflicts dialog
2	In the Table with Conflicts list, click on the entry that is likely to contain the lost information
3	Click on each entry in the Conflicting Records list When the lost information appears in the Conflict Details list, click on the Overwrite with Conflict button.
4	In the Conflicting Records list, you can also copy each Row ID to the clipboard by right-clicking on the row and selecting the Copy to Clipboard menu option

3.4.2 Configure User Security

User security in Enterprise Architect can be used to limit the access to update functions within the model.

Topics:

Topic	Detail	See also
User Security	Security in Enterprise Architect is a means of improving collaborative design and development by preventing concurrent editing and limiting the possibility of inadvertent model changes by users not designated as model authors User security is not enabled by default in Enterprise Architect; the Administrator must enable it first Where user security is enabled a password is required to log in to the model Elements can be locked per user or per group	Enable Security ^[196]
Security Policies	Enterprise Architect User Security offers two policies: <ul style="list-style-type: none"> • Standard security mode - all elements are initially unlocked and, as necessary, a user can set a user or group lock on any element or set of elements in order to make changes and protect those changes • Rigorous security mode - an Enterprise Architect model is read-only and nothing in the model can be edited until explicitly checked out with a user lock 	Security Policy ^[197]
Security Operations For Administrators	A number of security tasks can only be performed by users with Administrative rights to the model	Set Security Policy ^[197]

Topic	Detail	See also
		Enable Security ^[196] Maintain Users ^[198] Import User IDs From Active Directory ^[199] Change User Passwords ^[209] Assign User To Groups ^[201] View All User Permissions ^[202] Maintain Groups ^[203] View and Manage Locks ^[207] Password Encryption ^[208] Create Workflow Scripts ^[231]
Security Operations For Users	Other Security tasks can be performed by users who do not have Administrative rights	Lock Model Elements ^[210] Lock Packages ^[212] Apply a User Lock ^[213] Identify Who Has Locked An Object ^[215] Locked Element Indicators ^[214] Manage Your Own Locks ^[215] Change Your Own Password ^[209]
Security Scripts	With workflow administration permissions, you can also develop workflow scripts using the Scripting window Workflow scripts validate and control user input.	List of Available Permissions ^[205] Workflow Scripts ^[231] Scripting ^[1817] Workflow Script Functions ^[232]

Notes:

- User security can be enabled in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- For a number of operations in Enterprise Architect, if security is enabled a user must have the appropriate user or group access permission to perform the operation; however, if security is not enabled, the user does not have to have these access permissions

3.4.2.1 Enable Security

User security is not enabled by default in Enterprise Architect, the Administrator must enable it using an Authorization key obtained from the Sparx Systems website.

Access: [Project | Security | Enable Security](#)

Use to:

- Enable User Security on a project
- Disable User Security, in exactly the same way

How to:

Step	Action	See also
1	Obtain the Authorization Key from the Sparx Systems website on: <ul style="list-style-type: none"> • the Team Modeling Resources section (Trial User) or • the Registered Users section (Registered User; you must also have your Registered Users login and password) 	Team Modeling Resources web pages Registered Users web pages
2	In Enterprise Architect, select the Enable Security menu option The Enter authorization dialog displays	
3	In the Enter authorization key field, type the authorization key from the Sparx Systems website	
4	If required, select the Automatically apply Exclusive Edit Locks to diagrams checkbox This option is ignored in <i>Require User Lock</i> security mode. In standard (<i>User/Group Locking</i>) mode, this option blocks multiple users from simultaneously attempting to modify the same diagram (see <i>Notes</i> below)	Security Policy ^[197]
5	Click on the OK button Security is enabled, and an Admin user and user group are created with all access rights and a password of password	List of Available Permissions ^[205]
6	Select the Project Security Login as Another User menu option, and log in as Admin with the initial password of password If required, you can change the Admin password	Change Password ^[209]
7	Set up users and permissions as required.	Maintain Users ^[198] Assign User To Groups ^[201] Set Up Single Permissions ^[201] Maintain Groups ^[203] Set Group Permissions ^[204]

Notes:

- Once security has been enabled, you must have the **Security - Enable/Disable** access right to turn it off - the initial administrator automatically has this access right; the system prompts you to log off the project and log on again, but this is not strictly necessary
- If you re-enable security, be aware that any changes you have made to the **Admin** user (password and reduced access rights) are reset to password and full access
- The **Automatically apply Exclusive Edit Locks to diagrams** option is not displayed when disabling security, therefore to toggle the setting whilst security is enabled you must disable security and re-enable it; security settings (users, groups and permissions) and locks on elements, are NOT affected by this action
- If the **Automatically apply Exclusive Edit Locks to diagrams** option is selected, as a user modifies a diagram Enterprise Architect automatically applies a User Lock to the diagram, preventing any other user from modifying it
 It is creating difference between the database and buffer versions of the diagram that triggers the temporary lock, and elimination of difference that releases the lock; therefore, Enterprise Architect releases the lock when:
 - The user saves the changes to the diagram, with the **Save** icon or keyboard keys
 - The user undoes the last remaining action in the **Undo** list
 - The user saves or discards changes via the system prompt when they close the diagram
 If the diagram already has a User Lock or Group Lock that does not exclude the current user, this lock is set aside and saved when the temporary User Lock is applied. When the temporary User Lock is released, the pre-existing lock is restored

3.4.2.2 Security Policy

There are two possible security policies in Enterprise Architect.

Guide:

Topic	Detail	See also
User/Group Locking mode	<p>All elements and diagrams are considered unlocked and anyone can edit any part of the project</p> <p>However, when you edit a diagram, package or element, you lock the element or set of elements at either the user level or group level</p> <p>This mode is good for cooperative work groups where there is a solid understanding of who is working on which part of the model, and locking is used mainly to prevent further changes or to limit who has access to a part of the model</p>	
Require User Lock to edit mode	<p>More rigorous: the Enterprise Architect model is read-only, and everything is locked so that nobody can edit anything unless they explicitly check out the object with a user lock</p> <p>A single 'check out' function operates on a diagram to check out the diagram and all contained elements in one go</p> <p>There are also functions on the context (right-click) menus of packages, diagrams and elements in the Project Browser to apply a user lock when this mode is in use</p> <p>You would use this mode when there is a strict requirement to ensure only one person can edit a resource at one time; this is suitable for much larger projects where there might be less communication between users</p>	

Notes:

- Toggle between the security modes using **Project | Security | Require User Lock to Edit** - deselected

for **User/Group Locking** mode, selected for **Require User Lock to edit** mode

- When you add new elements in **User/Group Locking** mode (elements editable by default), no user lock is created automatically for the newly created element
- When you add new elements in **Require User Lock to Edit** mode (elements locked by default), a user lock is created on the new element to enable instant editing

3.4.2.3 Maintain Users

If you enable security you have access to the Security Users dialog, which you can use to set up more users for your model.

Access: [Project](#) | [Security](#) | [Manage Users](#)

Use to:

- Define the user IDs and initial passwords for the users that are to have access to the model

How to:

To set up a user for your model, follow the steps below:

Step	Action	See also
1	Select the Manage Users menu option The Security Users dialog displays, which you can use to set up new users by providing their name and other details You can also: <ul style="list-style-type: none"> • Import User IDs from a Windows Active Directory • Assign User IDs to User Groups • Set up single permissions for a user • View all permissions for the currently selected user 	Import User IDs from a Windows Active Directory ^[19] Assign User IDs to Groups ^[20] Single Permissions ^[20] View All ^[20]
2	To identify a new user on this dialog, click on the New button and type in the user's login ID, first name and last name If required, also provide the user's department name	
3	To set the user's password, click on the Change Password button The Change Password dialog displays	
4	In the New password field, type the user's password This must be 12 characters or less in length	
5	In the Retype new field, type the user's password again, for confirmation	
6	Click on the OK button <i>A Password Changed</i> message displays	
7	Click on the OK button	
8	When you have entered the details for the user, click on the Save button	
9	Click on either the: <ul style="list-style-type: none"> • New button to add another user, or • Close button to exit the Security Users dialog 	

Notes:

- You must have **Security - Manage Users** permission to maintain users, and **Change Password** permission to change the password of the current user; the initial **Admin** administrator automatically has these permissions
- You can transport the user definitions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options
- If you select the **Accept Windows Authentication** checkbox, when a user opens the model Enterprise Architect checks the user's database for their Windows ID and, if it matches, automatically logs the user in without prompting for a password
- The **Accept Windows Authentication** checkbox enables the **Import** button, which you can select to import user IDs from a Windows Active Directory
- As a security measure, the **Accept Windows Authentication** checkbox is automatically deselected if the project .eap file is moved to a different location; once the file has been relocated, you can select the checkbox again to apply Windows authentication from the new database

Learn More:

- [List of Available Permissions](#) ^[205]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

3.4.2.4 Import User IDs From Active Directory

You can import the Enterprise Architect security user IDs from Windows Active Directory.

When you import the user IDs, you should create an appropriate user group and assign the user IDs to that group; you can then assign appropriate permissions to the group.

Access: **Project | Security | Manage Users**

Use to:

- Import Windows user IDs as Enterprise Architect security user IDs for use with Windows Authentication

How to:

To import user IDs from Windows Active Directory, follow the steps below:

Step	Action	See also
1	On the Security Users dialog select the Accept Windows Authentication checkbox and click on the Import button The Import Users dialog displays	
2	Click on the down arrow in the Security Group field and select the appropriate security group for the imported user IDs	
3	Click on the Add button The Select Users screen displays	
4	Click on the Object Types button The Object Types dialog displays	

Step	Action	See also
5	Select the checkbox for the type of object to import from the Active Directory Click on the OK button to return to the Select Users dialog	
6	Click on the Locations button The Locations dialog displays	
7	Browse for and select the checkbox for the location to import from within the Active Directory Click on the OK button to return to the Select Users dialog	
8	In the Enter the object names to select field, either: <ul style="list-style-type: none"> type in the user IDs individually (click on the examples link to see examples of the correct formats) or click on the Advance button to search for IDs; the Select Users dialog redisplay with a Common Queries tab 	
9	In the Name and Description fields, type any characters or text that helps identify the IDs you are searching for	
10	In the Starts with field, click on the drop-down arrow and select the appropriate qualifier	
11	If required, select the Disabled accounts or Non-expiring password checkboxes, and/or select a value in the Days since last logon field, to further filter the IDs to search for	
12	Click on the Find Now button to initiate the search, and to display a list of IDs in the bottom panel of the dialog You can vary the types of information shown here by clicking on the Columns button and selecting the column headings to display	
13	When you have identified the IDs to import, click on a required ID (or press (Ctrl) or (Shift) while you click to select several) and click on the OK button The Select Users dialog redisplay, with the selected ID or IDs listed in the Enter the object names to select field	
14	Click on the OK button to redisplay the Import Users dialog with the selected users' names listed in the Users panel	
15	Click on the Import button to add the user IDs to the Security Users dialog	
16	Click on a user ID to populate the dialog fields with the user ID details, to set group membership and to set single permissions as required.	Group Membership ^[20] Set Single Permissions ^[20]

Notes:

- Enterprise Architect generates random passwords for Windows user IDs; however, if required you can assign a new password to an imported user ID

Learn More:

- [Maintain Groups](#) ^[20]
- [Change Password](#) ^[20]

3.4.2.5 Assign User To Groups

A user might belong to one or more work groups, each of which would have permissions to perform work in specific areas of Enterprise Architect.

Access: [Project | Security | Manage Users: Group Membership button](#)

Use to:

- Assign a user's ID to one or more work groups

How to:

To assign a user to user groups follow the steps below:

Step	Action	See also
1	On the Security Users dialog click on the Group Membership button The User Groups dialog displays	
2	Select the checkbox against each group this user belongs to	
3	Click on the OK button to assign the user to the selected groups	

Notes:

- To assign users to groups, you must have **Security - Manage Users** permission; the initial Admin administrator automatically has this permission.
- User groups must already exist in order to assign users to them

Learn More:

- [List of Available Permissions](#)^[205]
- [Maintain Groups](#)^[203]

3.4.2.6 Set Up Single Permissions

A user commonly belongs to one or more work groups that give that user group permissions to work in areas of Enterprise Architect. You can also assign additional, specific user permissions to that user ID.

Access: [Project | Security | Manage Users: Single Permissions button](#)

Use to:

- Define the specific permissions an individual user has to perform work in Enterprise Architect

How to:

To set up single permissions for a user follow the steps below:

Step	Action	See also
1	On the Security Users dialog, click on the Single Permissions button	

Step	Action	See also
	The User Permissions dialog displays	
2	Select the checkbox against each specific permission to assign to this user Click on the: <ul style="list-style-type: none"> Select All button to select all permissions for the user, or Deselect All button to clear all selected permissions 	
3	Click on the OK button to assign the selected permissions to the user	

Notes:

- You must have **Security - Manage Users** permission to assign permissions to users; the initial Admin administrator automatically has this permission
- A user's total permissions are those granted by Group Membership plus those granted by specific permission assignment
- You can transport the user permissions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options

Learn More:

- [List of Available Permissions](#)^[205]
- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

3.4.2.7 View All User Permissions

This topic explains how to list all the permissions a user has.

Access: **Project | Security | Manage Users**

Use to:

- Display a list of the permissions a specific user has, derived from their individual profile and from their membership of security groups

How to:

Step	Action	See also
1	Select the Manage Users menu option The Security Users dialog displays	
2	Select the required user Click on the View All button A dialog displays, listing all of the user's permissions	

3.4.2.8 Maintain Groups

This topic explains how to create a security group. Security groups make it easy to configure sets of permissions and apply them to a number of users in one action.

Access: [Project | Security | Manage Groups](#)

Use to:

- Create a security group prior to:
 - Assigning permissions to the group
 - Assigning users to the group

How to:

To set up a security group, follow the steps below:

Step	Action	See also
1	Select the Manage Groups menu option The Security Groups dialog displays	
2	In the Group Name and Description fields, type the security group name and a description of the group	
3	Click on the Save button	
4	<p>The group name can be selected as the recipient of an internal email; when an email is sent to the group, all members of the group receive that email in their Personal Information window</p> <p>However, the group name can act as either:</p> <ul style="list-style-type: none"> • A mail list, in which case each group member receives their own copy of the message, or • A mailbox, in which case the email is a single entity - the group members do not receive separate instances of it; therefore, if one group member responds to or deletes the email, the other group members see that action as if they had performed it themselves <p>To make the group name act as a mailbox, select the Shared Mail checkbox against the group name in the Groups panel</p> <p>To use the group name as a mail list, leave the checkbox unselected</p>	Internal Mail <small>[378]</small>

Notes:

- You must have **Security - Manage Users** permission to manage user groups; the initial Admin administrator automatically has this permission
- You do not define groups as group logins with passwords; if you intend to use a group login, you can define a single-user login and password that all group members use (that is, Enterprise Architect allows multiple logins under one user ID)
- Emails already sent to a group as a mail list and those sent to a group as a mailbox cannot be interchanged; if you change the status of the **Shared Mail** checkbox, the only way to change the distribution of past emails is to forward them to the group name again

Learn More:

- [Maintain Users](#) ^[198]
- [Set Group Permissions](#) ^[204]
- [Assign User To Groups](#) ^[201]
- [List of Available Permissions](#) ^[205]

3.4.2.9 Set Group Permissions

This topic explains how to assign permissions to a security group.

Access: **Project | Security | Manage Groups: Set Group Permissions button**

Use to:

- Assign permissions to a previously-created user group, to work in various areas of Enterprise Architect

How to:

To assign permissions to a security group, follow the steps below:

Step	Action	See also
1	Select the Manage Groups menu option The Security Groups dialog displays	
2	Click on the Set Group Permissions button The Group Permissions dialog displays	
3	Select the checkbox against each required permission Click on the: <ul style="list-style-type: none"> • Select All button to select all permissions • Deselect All button to clear all selected permissions 	
4	Click on the OK button to assign the permissions All of the users assigned to this group share in this set of permissions	

Notes:

- You can transport the group permissions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options
- You must have **Security - Manage Users** permission to assign permissions to user groups; the initial **Admin** administrator automatically has this permission

Learn More:

- [Maintain Groups](#) ^[203]
- [Assign User To Groups](#) ^[201]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]
- [List of Available Permissions](#) ^[205]

3.4.2.10 List of Available Permissions

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled, users can only perform certain tasks if they have the appropriate permissions for those tasks.

Some permissions take precedence over others. For example, if a user has **Use Version Control** permission, they can modify model elements on import even if they do not have **Update Element** permission.

Reference:

Permission	Enables the user to	See also
Administer Database	Compact and repair a project database	Compact a Project [418] Repair a Project [419]
Admin Workflow	Develop and manage workflow scripts	Workflow scripts [231]
Audit Settings	Change the audit settings in the Audit Settings dialog	Audit Settings [300]
Audit View	Enable auditing and display data in the Audit View and Audit History tab	Audit View Controls [304] Audit History Tab [307]
Baselines - Manage	Create, delete, import and export Baselines	Package Baselines [308]
Baselines - Restore	Merge data into the project model from a Baseline or XML file	Package Baselines [308]
Change Password	Change your own password or (Administrator) another user's password	Change Password [209]
Check Data Integrity	Check and repair project data integrity	Check Project Data Integrity [414]
Configure Datatypes	Add, modify and delete datatypes	Data Types [777]
Configure Images	Configure alternative element images	Using The Image Manager [593]
Configure Packages	Configure controlled packages and package properties	Configure Packages [329]
Configure Resources	Create and manage Resources window items: RTF templates, patterns, profiles and favorites	Resources [789]
Configure Stereotypes	Add, modify and delete Stereotypes	Stereotype Settings [773]
Configure Version Control	Set up version control options for the current model	Version Control Settings Dialog [270]
Export XML	Export a model to XML	Export to XML [320]
Generate Documents	Generate RTF and HTML documents from model packages	RTF Documents [1726] HTML Reports [1802]

Permission	Enables the user to	See also
Generate Source Code and DDL	Generate source code and DDL from a model element, and synchronize code against model elements if it already exists	Generate Source Code ^[1490] Generate DDL for a Table ^[1379] Synchronize Model and Code ^[1506]
Import XMI	Import a model from XMI	Import from XMI ^[323]
Lock Objects	Lock an element or package	Lock Model Elements ^[210] Lock Packages ^[212]
Manage Diagrams	Create new diagrams, copy and delete existing diagrams, and save a diagram as a UML Pattern	Create a Pattern ^[1022] Add New Diagrams ^[568] Copy a Diagram ^[583] Delete Diagram ^[580]
Manage Glossary	Create, edit and delete glossary items in the Project Glossary	Project Glossary ^[362]
Manage Issues	Update and delete model Issues	Project issues ^[358]
Manage Project Calendar	Add, update and delete Project Calendar events; those without this permission can view calendar items	Calendar ^[392]
Manage Project Information	Update and manage project resources, metrics and risks	Project Resources ^[348]
Manage Reference Data - Update	Update and delete reference items	Reference Data ^[772]
Manage Replicas	Create and synchronize replicas	Create Replicas ^[190] Synchronize Replicas ^[191]
Manage Tests	Update and delete Test records	The Testing Workspace ^[1695]
Reverse Engineer from DDL and Source Code	Reverse engineer from source code or ODBC, and synchronize model elements against code	Import Source Code ^[1508]
Security - Enable/Disable	Disable user security in Enterprise Architect	Enable Security ^[196]
Security - Manage Locks	View and delete element locks	View and Manage Locks ^[207]
Security - Manage Users	Maintain users, groups and assigned permissions	Maintain Users ^[198] Assign User to Groups ^[201] Set Up Single Permissions ^[201]

Permission	Enables the user to	See also
Spell Check	Spell check package and set spell check language	Using the Spell Checker ^[376]
Transfer Data	Transfer model between different repositories	Perform a Project Data Transfer ^[343]
Transform Package	Perform transformations of packages and elements	Model Transformations ^[1307]
Update Diagrams	Update diagram appearance, properties and layout, including the Page Setup dialog	Diagram Properties ^[569] Set Diagram Page Size ^[600]
Update Element	Save model changes (including delete) for elements, packages, and relationships	Element Tasks ^[625] Packages ^[533] Connector Tasks ^[740]
Use Version Control	Check files in and out using version control	Check In and Check Out Packages ^[275]

3.4.2.11 View and Manage Locks

Enterprise Architect provides a function to view and manage active locks on elements.

Access: [Project](#) | [Security](#) | [Manage Locks](#)

Use to:

- Periodically examine and, if necessary, delete locks placed on elements by users

How to:

To view locks and, if necessary, delete them, follow the steps below:

Step	Action	See also
1	Select the Manage Locks menu option The Active Locks dialog displays	
2	In the View Locks For panel, click on the radio button for the type of lock to view <ul style="list-style-type: none"> All Groups Only Users Only Locks of the selected type are listed in the Active Locks panel	
3	To remove a lock, click on it and click on the Unlock Selected button	
4	When finished, click on the Close button to close the dialog	

Notes:

- You must have **Security - Manage Locks** permission to view and delete user locks; the initial **Admin** administrator automatically has this permission
- If you want to display the resulting information in a more readable layout, you can resize the dialog and its columns
- If an element is locked then connectors attached to it are also locked, so to unlock the connector you must unlock the element; however, under certain circumstances you can add new connectors to a locked element

Learn More:

- [List of Available Permissions](#)^[205]
- [Add Connectors to Locked Elements](#)^[211]

3.4.2.12 Password Encryption

This topic is retained to support regression to releases of Enterprise Architect prior to version 7.1; for password encryption for all repositories at and beyond release 7.1, you create model shortcuts. Enterprise Architect Security Administrators of projects on SQL Server or Oracle repositories can encrypt the password used to set up the connection between Enterprise Architect and the repository. The Enterprise Architect user does not have the real password, thereby preventing them from accessing the repository using other tools such as Query Analyzer or SQLPlus.

Access: **Project | Security | Encrypt Password**

Use to:

- Encrypt the password used to set up the connection between Enterprise Architect and a SQL Server or Oracle repository

How to:

To encrypt a repository password, follow the steps below:

Step	Action	See also
1	Select the Encrypt Password menu option A simple dialog displays	
2	In the Password field, enter the password to access the repository The Encrypted field displays the modified password to be used by users	
3	Instruct users to connect Enterprise Architect to the repository using the encrypted password prefixed with \$\$	<ul style="list-style-type: none"> • Connect to Oracle Data Repository^[164] • Connect to SQL Server Data Repository^[163]

Notes:

- Do not use the **Test Connection** button as it can cause an error with encrypted passwords
- For SQL Server repositories, you must enter the **Initial Catalog** details from the All tab of the Data Link Properties dialog

Learn More:

- [Model Shortcuts](#)^[140]

3.4.2.13 Change Password

This topic explains how to change a password when security is set.

Access: **Project | Security | Change Password** (User)
Project | Security | Manage Users (Administrator)

Use to:

- Change your own password (User)
- Set or change any user's password (Administrator)

How to:

To change your own user password, follow the steps below:

Step	Action	See also
1	Select the Change Password menu option The Change Password dialog displays	
2	In the Enter old password field, type your current password	
3	In the New password field, type your new password This must be 12 characters or less in length	
4	In the Retype new field, type your new password again, for confirmation	
5	Click on the OK button The <i>Password Changed</i> message displays	
6	Click on the OK button to clear the message Your new password is effective next time you log in	

To set or change any user's password as Administrator, follow the steps below

Step	Action	See also
1	Select the Manage Users menu option The Security Users dialog displays	
2	In the Users: panel, click on the user's name The user's details display in the dialog fields	
3	Click on the Change Password button The Change Password dialog displays	

Step	Action	See also
4	In the New password field, type the user's password This must be 12 characters or less in length You do not have to enter the user's current password	
5	In the Retype new field, type the user's password again, for confirmation	
6	Click on the OK button The <i>Password Changed</i> message displays	
7	Click on the OK button	

Notes:

- A user must have **Change Password** permission to change a password; the initial Admin administrator automatically has this permission

Learn More:

- [List of Available Permissions](#) ^[205]

3.4.2.14 Lock Model Elements

This topic explains how to set a lock on a model object in **User/Group Locking** mode.

In **Require User Lock to edit** mode, the *Apply a User* lock process applies.

Access:

Select:

- **Project Browser | Package context menu | Lock**
- **Project Browser | Diagram context menu | Lock**
- **Project Browser | Element context menu | Lock**
- **Diagram background context menu | Lock** or
- **Diagram Element context menu | Lock**

Use to:

- Set an editing lock on a modeling object so that:
 - Nobody can edit the object
 - Only you can edit the object, or
 - Only a member of your user group can edit the object
- Clear any editing lock on the object

How to:

To set a lock on an object in **User/Group Locking** mode, follow the steps below:

Step	Action	See also
1	Select the Lock option on the appropriate context menu	

Step	Action	See also
	The Element Lock dialog displays	
2	Select the radio button for the required option: <ul style="list-style-type: none"> • No lock - do not lock this object; clear any existing lock • Full lock - lock this object so that no-one can edit it • User lock - lock this object so that only the locking user can make further edits • Group lock - lock this element so that any member of the specified group can edit the object, but others are excluded 	
3	If you have selected the Group lock option, in the GroupID field click on the drop-down arrow and select the group that can edit the object	
4	Click on the OK button	

Notes:

- You must have **Lock Objects** permission to lock an element
- If the item already has a lock, only the corresponding lock option and the **No lock** option are shown; you have to release the lock in order to set a different type of lock
- If a diagram is locked and you select an object on it, the object border displays in red, indicating that you cannot change the object

Learn More:

- [List of Available Permissions](#)^[205]
- [Security Policy](#)^[197]
- [Apply a User Lock](#)^[213]

3.4.2.15 Add Connectors To Locked Elements

When working in a diagram containing locked elements, you can add connectors depending on the lock status of the source and target elements.

Reference:

Element Status	Add Connectors	See also
Source unlocked, target unlocked	Yes, any kind of connector can be added	
Source unlocked, target locked	Yes, except for Composition connectors	
Source locked, target unlocked	No, except for Composition connectors	
Source locked, target locked	No, prohibited for all connectors	

Notes:

- Generally, a connector can be added if its source is unlocked - you are modifying what the source can see
- The exception is Composition connectors, where the target (the parent) must be unlocked - you are modifying the parent by adding children
- Connectors with locked source or target elements are also locked; to unlock the connector, you must

unlock the source and/or target element

Learn More:

- [View and Manage Locks](#) ^[207]

3.4.2.16 Lock Packages

This topic explains how to lock the complete contents of a package (and optionally all contents of its child packages) in **User/Group locking** mode, in one operation.

Access: **Project Browser | Package context menu | Lock Package**

Use to:

- Automatically set or clear locks on elements and diagrams within a package in **User/Group locking** mode, as if the locks had been set or cleared individually

How to:

To lock a package, follow the steps below:

Step	Action	See also
1	Deselect the Project Security Require User Lock to Edit menu option	Security Policy ^[197]
2	In the Project Browser, right-click on the package to lock The context menu displays	
3	Select the Lock Package menu option The Lock/Unlock Package(s) dialog displays	
4	In the Lock Type panel, select the appropriate radio button for the lock to apply, as for an individual object lock	Lock Model Elements ^[210]
5	As required, select the checkboxes to lock: <ul style="list-style-type: none"> • Elements and/or diagrams • The contents of child packages (that is, lock the whole branch) 	
6	Click on the OK button to apply the lock.	

Notes:

- If security is enabled you must have **Lock Objects** permission to lock a package

Learn More:

- [List of Available Permissions](#) ^[205]

3.4.2.17 Apply a User Lock

In the **Require User Lock to Edit** security mode, where a User Lock is required before any edit can occur, you can set or release the lock in either a diagram or the **Project Browser**.

Access:

Select:

- **Project Browser | Package context menu | Apply/Release User Lock**
- **Project Browser | Diagram context menu | Apply/Release User Lock**
- **Project Browser | Element context menu | Apply/Release User Lock**
- **Diagram background context menu | Apply/Release User Lock** or
- **Diagram Element context menu | Apply/Release User Lock**

Use to:

- Set an editing lock on a modeling object so that only you can edit the object
- Clear the editing lock on the object

How to:

To set or clear a user lock in **Require User Lock to Edit** security mode, follow the steps below:

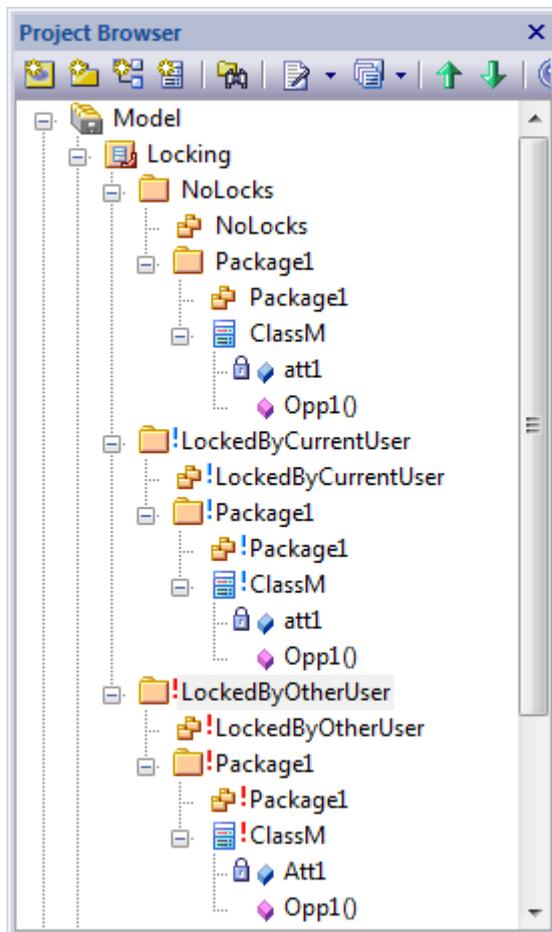
Step	Action	See also
1	Right-click on the required object in the Project Browser or diagram, and select the Apply/Release User Lock option	
2	Select the appropriate radio button to apply a user lock to, or release a user lock on, the selected object Enterprise Architect adjusts the lock for the element, or for the diagram and any elements contained in the diagram.	
3	For a package, if required, select to also lock all child packages If any elements in the package tree are locked by other users, a list of elements that couldn't be locked displays	

Learn More:

- [Security Policy](#)^[197]

3.4.2.18 Locked Element Indicators

When an item is locked through User Security, the lock is indicated in the Project Browser by a marker against the item, as shown below:



Guide:

The meanings of the markers depend on the security mode:

Security Mode	Marker	Meaning	See also
Require User Lock to Edit	No marker	There is no lock, the item is not editable, but any user can now apply a user lock to edit the item	Security Policy [197] Apply a User Lock [213]
	Blue exclamation mark	The current user has applied a user lock and can edit the item; no other user can edit the item	
	Red exclamation mark	Another user has applied a user lock, and the current user cannot edit the item You can find out which user has locked the item	Identify Who Has Locked An Object [215]

Security Mode	Marker	Meaning	See also
User/Group Lock	No marker	There is no lock, the item is editable, but any user can now apply a user or group lock	Security Policy [197] Lock Model Elements [210]
	Blue exclamation mark	The item has a lock set by the current user or a group having the current user as a member, and the user can edit the item	
	Red exclamation mark	The item has a lock set by another user, or a group of which the current user is not a member; the current user cannot edit the item You can find out which user has locked the item	Identify Who Has Locked An Object [215]

Notes:

- If a diagram is locked and you select an object on it, the object border displays in red; this indicates that you cannot change the object

3.4.2.19 Identify Who Has Locked An Object

If you find that a diagram, package or element is locked, you can find out which group or user currently holds the lock on that item.

Access:

- **Project Browser | Package context menu | Lock**
- **Project Browser | Diagram context menu | Lock**
- **Project Browser | Element context menu | Lock**

Use to:

- Determine which user or user group has locked the package, diagram or element in which you are interested

How to:

To identify the lock holder, follow the steps below:

Step	Action	See also
1	In the Project Browser, right-click on the diagram, package or element that is locked by another user or user group, and select the Lock option A message box displays showing which group or user currently holds the lock on that item	

3.4.2.20 Manage Your Own Locks

You can view and delete your own user-level locks in Enterprise Architect. This is especially useful when working in *Require user locks to edit* security mode.

Access: **Project | Security | My Locks**

Use to:

- Review the locks you have set and, where necessary, clear them

How to:

To manage your locks, follow the steps below:

Step	Action	See also
1	Select the My Locks menu option The My Locks dialog displays, showing all the locks you have set	Lock Model Elements ^[210] Apply a User Lock ^[213] Lock Packages ^[212]
2	If you intend to clear locks, either: <ul style="list-style-type: none"> • Select a lock you intend to clear • Press (Ctrl) or (Shift) as you select a number of locks or a range of locks respectively, to clear • Click on Select All to select all of your locks, or • Click on Select None to clear your selection, if you have made an error 	
3	If you have selected locks to clear, click on the Unlock Selected button The objects that were locked are now unlocked	

Learn More:

- [Security Policy](#) ^[197]

3.4.3 Team Review Tools

The Enterprise Architect Project *Team Review* can be used by the team members to discuss the development and progress of a project.

Access: [View | Team Review](#) (Ctrl+Alt+U)

Topics:

Topic	Detail	See also
Team Review Facility	The Team Review is discussion forum within your development team community A point of discussion and the responses to that point are created as <i>Posts</i> Posts are held in <i>Topic</i> groups, which are in turn organized into <i>Categories</i> ; your organization decides on the grouping and organization of Posts	Add a new Category ^[220] Add a New Topic ^[221] Add a New Post ^[222]

Topic	Detail	See also
	<p>You can create text to explain Categories and Topics, and you can create and respond to Posts and replies</p>	
<p>Displays</p>	<ul style="list-style-type: none"> The <i>Team Review window</i>, used to create new Categories and Topics and to delete messages, using a context menu <p>Each item in the hierarchy displays a mouse-over tooltip, showing the item title, the author's name and the date and time the item was created</p> <ul style="list-style-type: none"> The <i>Team Review tab</i>, in the main work area, used to view, print, create, edit, and reply to review Posts <p>This tab has a status bar that shows the item author's name, the date and time the item was created, and the date and time the item was last modified</p> <ul style="list-style-type: none"> The <i>linked objects panel</i>, located at the bottom of the Team Review tab, used to associate model elements, diagrams, external files and other postings with a review posting 	<p>The Team Review Context menu [218]</p> <p>Reply to a Post [224]</p> <p>Add Object Links [227]</p>
<p>Features</p>	<ul style="list-style-type: none"> Display or edit the text on the Team Review tab by clicking on the item name in the Team Review window Using the Search icon in the Team Review toolbar, search for text strings in the item titles to locate posts on a specific topic Add resources to a Category, Topic or Post; resources such as diagram images and XML files of packages, are held in a <i>Resources</i> folder under the selected Team Review item Change the loading behavior of the forum Access team reviews from other projects, including those on servers 	<p>Edit an Item [226]</p> <p>Search Team Review [229]</p> <p>Team Review Resources [228]</p> <p>Team Review Options [229]</p> <p>Team Review Connections [230]</p>
<p>Icons</p>	<p>The icons beside the review items have the following meanings:</p> <ul style="list-style-type: none">  Post read  Post unread  Reply read  Reply unread  Category (name in bold indicates category is unread)  Topic read  Topic unread  <i>Resources</i> folder for a Category, Topic or Post  Diagram or clipboard image within <i>Resources</i> folder  XML File of package, within <i>Resources</i> folder. 	

3.4.3.1 Context Menu

The **Team Review** context menu enables you to access a wide range of functions, depending on the object that you right-click on.

Action	Usage	Shortcut	See also
New Category New Topic New Post	<p>Add a new Category, new Topic or new Post to the Team Review</p> <p>Alternatively, for a Topic or Post, click on the New icon in the window toolbar</p>		Add a New Category Add a New Topic Add a New Post
New Category From Template New Topic from Template New Post from Template	<p>Add a new Category, new Topic or new Post, based on a defined template</p>		Add a New Category Add a New Topic Add a New Post
Post Reply	<p>Create a reply to the selected Post</p> <p>Alternatively, click on the New icon in the window toolbar</p>		Reply to a Post
Rename	<p>Edit the name, in situ, of the currently-selected item</p>	F2	
Copy Path to Clipboard	<p>Copy the path of the currently-selected item to the clipboard</p> <p>You can then paste the path into a document or file to add the discussion to any text concerning the item</p>	Ctrl+C	
Show Contents	<p>Display the description or text of the selected item in the Team Review window, if the Team Review tab is not yet open</p>		
Share Resource	<p>(If anything other than a Resources folder is selected)</p> <p>Add a package from the current model, or an image of the currently-displayed diagram, or an image from the clipboard, to the <i>Resources</i> folder under the selected Category, Topic or Post</p> <p>If the <i>Resources</i> folder does not exist, this option creates it</p>		Team Review Resources
Add Package From Current Model	<p>(If a Resources folder is selected)</p> <p>Export a package as an XML file from the current model as a resource of the selected Category, Topic or Post</p> <p>You browse for and select the required package using the Select <item> dialog</p>		Team Review Resources Select <item> Dialog
Import to Current Model	<p>(If a package XML file resource is selected)</p> <p>Import the resource package to the current model</p> <p>You browse for and select the required target</p>		Select <item> Dialog

Action	Usage	Shortcut	See also
	<p>package using the Select <item> dialog; the resource is imported as a child of that package</p> <p>This is a useful option for transferring relevant packages from the Team Review of one model into another model</p>		
Add Image of Active Diagram	<p>(If a Resources folder is selected)</p> <p>Add an image of the currently-displayed diagram as a resource of the selected Category, Topic or Post</p> <p>You are prompted to provide a reference name for this image</p>		Team Review Resources ^[228]
Add Active Profiler Report	<p>(If a Resources folder is selected)</p> <p>Add an active Profiler Report as a resource of the selected Category, Topic or Post</p> <p>You are prompted to browse for and select the appropriate active report</p>		Team Review Resources ^[228] Save Report in Team Review ^[1662]
Image From Clipboard	<p>(If a Resources folder is selected)</p> <p>Add an image held on the clipboard as a resource of the selected Category, Topic or Post</p>		Team Review Resources ^[228]
View Image	<p>(If a diagram image resource is selected)</p> <p>Open the View Image window, containing an image of the selected diagram</p> <p>Alternatively, double-click on the image name</p>		Team Review Resources ^[228]
Copy Image To Clipboard	<p>(If an image resource is selected)</p> <p>Copy the image or diagram image to the clipboard</p>		Team Review Resources ^[228]
Refresh Category 'xyz' Refresh Topic 'xyz' Refresh Post 'xyz'	<p>Refresh the currently-open Category, Topic or Post, getting new replies, Posts and Topics that other users might have created</p> <p>However, if you open another Category, Topic or Post the Team Review always displays the latest information from the database</p> <p>Alternatively, click on the Refresh icon in the window toolbar</p>		
Reload Current Connection	Reload the entire Team Review connection, getting new Categories, Posts and Topics		
Review Status	<p>Assign or clear a status marker against the selected Category, Topic or Post; you can mark the item as:</p> <ul style="list-style-type: none"> • Awaiting Approval • Approved • Rejected <p>Or clear the marker (None)</p>		

Action	Usage	Shortcut	See also
Mark	See the Mark submenu, below		
Connections...	<p>Access other Team Reviews from other Enterprise Architect models or models located on servers</p> <p>Alternatively, click on the drop-down arrow in the Connection Options field in the window toolbar, and select one of the listed models</p> <p>The <Configure Connections> option enables you to add and connect to additional Team Reviews</p>		Team Review Connections [230]
Options...	Change the loading behavior of the Team Review		Team Review Options [229]
Delete Category <xyz> Delete Topic <xyz> Delete Post <xyz> Delete Resource <xyz>	Delete this Category, Topic, Post or reply and all sub-topics and sub-posts, or delete the resource attached to the item		Edit an Item [226]

The Mark Submenu:

Action	Usage	Shortcut	See also
All items as Read	Mark the entire contents of the Team Review as read [1007]		Team Review Tools [216]
All items as Unread	Mark the entire contents of the Team Review as unread [417]		Team Review Tools [216]
Branch as Read	Mark this item and all its contents as read		
Branch as Unread	Mark this item and all its contents as unread		
'xyz' as Unread	Mark only this item as unread		

3.4.3.2 Add a New Category

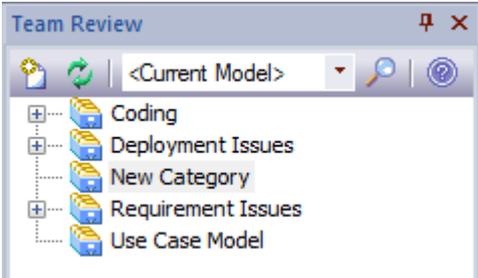
This topic explains how to add new Categories to the Team Review.

Access: **View | Team Review** (Ctrl+Alt+U)

Use to:

- Create a new Category in which to store related Topics and Posts

How to:

Step	Action	See also
1	<p>Right-click on a blank area in the Team Review window and select the New Category context menu option</p> <p>A new Category icon displays in the hierarchy</p> 	Context Menu ^[218]
2	<p>Overtyping the <i>New Category</i> text with the name of the Category, and clicking off the name</p> <p>The Category description editor displays in the Team Review tab; type the appropriate Category description, if required</p>	
3	Add new Topics and/or resources to the Category	Add a New Topic ^[221] Team Review Resources ^[228]
Alternatively		
1	<p>Right-click on a blank area in the Team Review window and select the New Category from template context menu option</p> <p>A new Category icon displays in the hierarchy</p>	Context Menu ^[218]
2	<p>Overtyping the <i>New Category</i> text with the name of the Category, and clicking off the name</p> <p>The Create New Category dialog displays</p>	
3	<p>Click on the Copy template drop-down arrow and select a predefined template for the Category description</p> <p>Click on the OK button</p>	
4	The Category description editor displays in the Team Review tab; type the appropriate Category description, if required	
5	Add new Topics and/or resources to the Category	Add a New Topic ^[221] Team Review Resources ^[228]

3.4.3.3 Add a New Topic

This topic explains how to add new Topics to a Category in the Team Review.

Access: [View | Team Review](#) (Ctrl+Alt+U)

Use to:

- Create a new Topic in which to store related Posts and their responses

How to:

Step	Action	See also
1	Either: <ul style="list-style-type: none"> Right-click on the required Category name in the Team Review window and select the New Topic context menu option Click on the Category name and click on the New icon in the Team Review toolbar, or Click on the Category name and press (Ctrl+N) A new Topic icon displays under the Category name	Context Menu ^[218]
2	Overtyping the <i>New Topic</i> text with the name of the Topic, and clicking off the name The Topic description editor displays in the Team Review tab; type the appropriate Topic description, if required	
3	Add new Posts and/or resources to the Topic	Add a New Post ^[222] Team Review Resources ^[228]
Alternatively		
1	Right-click on the required Category name in the Team Review window and select the New Topic from template context menu option A new Topic icon displays under the Category Name	Context Menu ^[218]
2	Overtyping the <i>New Topic</i> text with the name of the Topic, and clicking off the name The Create New Topic dialog displays	
3	Click on the Copy template drop-down arrow and select a predefined template for the Topic description Click on the OK button	
4	The Topic description editor displays in the Team Review tab; type the appropriate Topic description, if required	
5	Add new Posts and/or resources to the Topic	Add a New Post ^[222] Team Review Resources ^[228]

3.4.3.4 Add a New Post

This topic explains how to create a new Post on a Topic in the **Team Review**.

Access: **View | Team Review** (Ctrl+Alt+U)

Use to:

- Create a blank Post
- Create a Post based on a predefined template

- Create a Post from a file link

How to - Create a blank message:

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> • Right-click on the appropriate Topic name in the Team Review window and select the New Post context menu option • Click on the Topic name and on the New icon in the Team Review toolbar, or • Click on the Topic name and press (Ctrl+N) <p>A new Post icon displays underneath the Topic name</p>	Context Menu ^[218]
2	<p>Overtyping the <i>New Post</i> text with the name of the Post, and clicking off the name</p> <p>The Post text editor displays in the Team Review tab</p>	
3	Type the text of the Post as required	Edit an Item ^[226]

How to - Create a Post based on a predefined template:

Step	Action	See also
1	<p>Right-click on the appropriate Topic name in the Team Review window and select the New Post from template context menu option</p> <p>A new Post icon displays in the hierarchy</p>	Context Menu ^[218]
2	<p>Overtyping the <i>New Post</i> text with the name of the Post, and clicking off the name</p> <p>The Create New Post dialog displays</p>	
3	Click on the drop-down arrow in the Copy template field, and select a predefined template for the Post contents	
4	<p>Click on the OK button</p> <p>The template structure is displayed in the Team Review tab</p>	
5	Enter the text of the Post as required	Edit an Item ^[226]

How to - Create a Post based on a link to an external file:

Step	Action	See also
1	Open a file browser (such as Windows Explorer) and locate the file	
2	<p>Click on the file name and drag it onto the appropriate Topic name in the Team Review window</p> <p>A new Post icon is created underneath the selected Topic, and the body of the Post is shown in the Team Review tab</p>	

Step	Action	See also
	A link to the source file is created at the start of the message	
3	Click twice on the <i>New Post</i> text in the Team Review window, and overtype the text with the name of the Post	
4	In the Team Review tab, create and edit text around the file link, and add further links if required	Edit an Item [226]

When you have created the Post:

Topic	Detail	See also
You can create links from the Post	To elements and diagrams from the: <ul style="list-style-type: none"> • Project Browser • Model Search • Model Views window • Package Browser 	Add Object Links [227] The Project Browser [44] Model Search [475] Model Views [464] Package Browser [456]
	To related Team Review: <ul style="list-style-type: none"> • Categories • Topics • Posts 	Add Object Links [227]
	To external files, either in the text of the Post or in the Linked Objects panel	Add Object Links [227]
Add resources to the Post		Team Review Resources [228]
Other users can reply to it		Reply to a Post [224]

3.4.3.5 Reply to a Post

This topic explains how to reply to a Post in the **Team Review**.

Access: [View | Team Review](#) (Ctrl+Alt+U)

Use to:

- Create a response to a Post
- Create a response to an existing response to a Post

How To:

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> • Right-click on the required Post (or response) in the Team Review window and select the Post Reply context menu option • Click on the Post name and on the New icon in the Team Review toolbar, or • Click on the Post name and press (Ctrl+N) <p>A <i>Re:<Postname></i> entry displays underneath the Post you are replying to, and the cursor becomes active in the Team Review tab to enable you to create and edit your response</p>	Context Menu ^[218]
2	Type in, format and save the contents of the reply	Edit an Item ^[226]
Alternatively		
1	Open a file browser (such as Windows Explorer) and locate the file	
2	<p>Click on the file name and drag it into the Team Review window, over the Post to which you are replying</p> <p>A <i>Re:<Postname></i> entry displays underneath the Post, and the cursor becomes active in the Team Review tab to enable you to edit your response</p> <p>A link to the source file is created at the start of the message</p>	
3	Type in, format and save the contents of the reply	Edit an Item ^[226]

When you have created the Post:

Topic	Detail	See also
You can create links from the Post	To elements and diagrams from the: <ul style="list-style-type: none"> • Project Browser • Model Search • Model Views window • Package Browser 	Add Object Links ^[227] The Project Browser ^[447] Model Search ^[475] Model Views ^[464] Package Browser ^[456]
	To related Team Review: <ul style="list-style-type: none"> • Categories • Topics • Posts 	Add Object Links ^[227]
	To external files, either in the text of the Post or in the Linked Objects panel	Add Object Links ^[227]
Add resources to the Post		Team Review Resources ^[228]
Other users can reply to it		Reply to a Post ^[224]

3.4.3.6 Edit an Item

This topic explains how to edit or delete a Category, Topic, Post or reply.

How to:

Topic	Detail	See also
Select an item to edit	Click on the item name in the Team Review window The item text displays in the Team Review tab to enable you to edit the text	
Change the name of the item	Click on the item name in the Team Review window and press (F2) Overtyping the current name	
Delete an item	Either: <ul style="list-style-type: none"> Right-click on the item in the Team Review window and select the Delete <item> <name> context menu option, or Click on the item and press (Delete) A confirmation dialog displays; click on the Yes button to remove the item and any dependent items from the Team Review	Context Menu ^[218]
Edit the text	Access the Team Review Editor standard functions by right-clicking on the text, to display a hierarchy of context menus When you have completed your editing, select Right-click File Save and then click on another item in the Team Review window to exit the message The text is saved in the Team Review item To redisplay the text in the Team Review tab, click on the item name once	<ul style="list-style-type: none"> Scroll Through Text ^[1744] File and Print Options ^[1745] Cut and Paste Options ^[1746] Image and Object Imports ^[1749] Character Formatting ^[1750] Paragraph Formatting ^[1751] Tab Support ^[1753] Page Breaks and Repagination ^[1754] Insert Headers and Footers ^[1754] Insert Bookmarks ^[1754] Table Commands ^[1756] Sections and Columns ^[1759] Stylesheets and Table of Contents ^[1759]

Topic	Detail	See also
		<ul style="list-style-type: none"> • Text/Picture Frame and Drawing Objects ^[1763] • Search/Replace Commands ^[1764] • Hyperlink From Linked Document ^[732] • Create Elements From Linked Documents ^[732]

3.4.3.7 Add Object Links

In the **Team Review** tab you can create hyperlinks to elements and diagrams that are associated with a Post.

Use to:

- Rapidly navigate to the objects in the Project Browser
- Access the element properties
- Open a diagram directly from the Team Review
- Create links to:
 - Other Categories, Topics and Posts in the Team Review window
 - External files from a file browser

How to:

Topic	Detail	See also
Associate an element, diagram or Team Review item with a message	Drag the object into the Linked Elements panel at the bottom of the Team Review tab, from the: <ul style="list-style-type: none"> • Project Browser • Package Browser • Model Views window • Model Search dialog or • Team Review window 	The Project Browser ^[441] Model Search ^[475] Model Views ^[464] Package Browser ^[456]
Associate an external file with the message	Click on and drag the file name from any browser into either: <ul style="list-style-type: none"> • the linked elements panel or • the text of the message itself The filename becomes a link to the file; click on it to display the contents of the file The external file name also becomes a link to the file within the message when you drag the filename onto a Topic to create a Post	Add a New Post ^[222]

To access the navigation options of each object in the Linked Elements panel (outlined below) right-click on

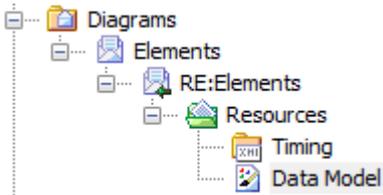
the object to display the navigation context menu.

Action	Usage	Shortcut	See also
Open	Open the diagram or external file		
Properties	Display the element properties for the selected element		
Find in all Diagrams	Open the diagram in which the element is used, or display a list of several diagrams in which the element has been used		
Delete Link	Delete the association between the message and the object		

3.4.3.8 Team Review Resources

You can add resources to a Category, Topic or Post within the **Team Review** window.

Topics:

Topic	Detail	See also
Resources	<ul style="list-style-type: none"> • XML files of packages within the current project • Active Profiler reports • Images of currently-active diagrams 	Save and Load Reports ^[1662] XML Import and Export ^[319]
Location	<p>Resources are created in a <i>Resources</i> folder underneath the selected Category, Topic or Post, as illustrated below:</p> 	
Management	<p>All Team Review resource management tasks are performed using options on the Team Review context menu</p> <p>You create the <i>Resources</i> folder by creating a resource underneath the selected Category, Topic or Post</p> <p>Similarly, you delete the <i>Resources</i> folder by deleting the last resource within it</p> <p>Having added a resource, you can:</p> <ul style="list-style-type: none"> • Reimport the package XML files to the model or • Display the diagram images 	Context Menu ^[218]

3.4.3.9 Search Team Review

The Team Review provides the facility to search the *titles* of all Team Review items.

Use to:

- Locate items referring to a specific subject

How to:

To search the Team Review, follow the steps below:

Step	Action	See also
1	In the Team Review window toolbar, click on the Search icon The search panel displays underneath the toolbar	
2	In the blank field, type the text string to search for	
3	If required, select the Match Case checkbox to locate text with the same case as the search string	
4	If required, select the Match Whole Word checkbox to locate only complete words that match the search string	
5	Click on the Find button The search locates the first instance of the search string in the title of a Category, Topic, Post or reply item, and displays the contents of that item in the Team Review tab	
6	To locate further instances of the text string, click on the Find button again	
7	To close the search panel, click on the Search icon in the toolbar again	

3.4.3.10 Team Review Options

You can change the loading behavior of the Team Review, using the Team Review Server Options dialog.

Access: **Right-click Team Review window | Options**

Action	Usage	Shortcut	See also
Mark All Unread	Resets all posts you have read to 'unread' (bolds all items in the Team Review)		
Load item data when required	The fastest loading option; Team Review data is only loaded on demand - for example, when you read a post		
Preload all data	Caches the entire contents of the Team Review on load; this takes longer to load but, once completed, navigating the Team Review is faster		

3.4.3.11 Team Review Connections

The **Connections** option enables you to access other Team Reviews from other Enterprise Architect models, including models located on servers.

Access: Through the Team Review Toolbar
From the Team Review context menu options

How to:

To switch to another Team Review, follow the steps below:

Step	Action	See also
1	<p>In the Team Review toolbar, click on the drop-down arrow in the Connection Options field and select the appropriate model name from the list, to connect to the Team Review for that model</p> <p>If the required model is not listed, click on the <Configure Connections> option; the Team Review Server Connections dialog displays</p> <p>Go to Step 3</p>	
2	<p><i>Alternatively:</i></p> <p>Right-click anywhere in the Team Review window and select the Connections context menu option; The Team Review Server Connections dialog displays</p> <p>In the list in the Connections panel, select the check box against the appropriate model name to connect to the Team Review for that model, and click on the Open button</p> <p>The connection now switches to the Team Review in the selected model</p> <p>If the required model is not listed, go to Step 3</p>	
3	<p>Select the appropriate Connection Type radio button and click on the New button</p> <ul style="list-style-type: none"> • For a .eap file, a browser dialog displays through which you can search for and select the appropriate model • For a model in a DBMS data repository, the Microsoft Data Link dialogs display, which enable you to locate and connect to the repository 	
4	<p>When you have selected and opened or connected to the required Enterprise Architect model, and returned to the Team Review Server Connections dialog, the model name displays in the Connection Name field and in the Connections panel</p>	
5	<p>Select the check box against the model name and click on the Open button to connect to the Team Review for that model</p> <p>The Team Review now shows the discussion in the selected model</p>	

For further details of the fields and buttons on the Team Review Server Connections dialog, refer to the table below.

Action	Usage	Shortcut	See also
Connection Name	Verify the name of the selected model		
Connection Type	Specify the type of Enterprise Architect model: a local .eap file (as above) or a model on a remote server		Connect to a Data Repository ^[16]
Target Model	Verify the path to the selected model		
New	Create a new Team Review connection		
Delete	Delete the currently selected connection from the Connections list		
Connections	List all Team Review connections created; click on the checkbox against the required connection		
Open	Switch the Team Review to the one selected in the Connections list		

3.4.4 Workflow Scripts

Enterprise Architect enables you to create workflow scripts that provide a robust approach to applying company policy and strengthening project development guidelines, by validating against the policy and procedures within the model itself.

Topic	Detail	See also
Project Governance	<p>Good corporate governance relies on well written and transparent project development guidelines and company policy.</p> <p>A project might be compromised if the appropriate policies and procedures are poorly understood and not followed correctly - effective governance can be hampered by human error and the costs of recovering from the inadequate compliance of developers.</p>	
Policies, Procedures and Development	<p>Company policy and procedures can be integrated with the development process to manage work flows, determine access rights, extend role based security permissions and respond to property change events.</p> <p>This approach reduces compliance costs, enhances collaborative development and gives you confidence that projects are being developed correctly the first time around.</p> <p>Development teams can adhere to best practice guidelines that govern model validation, change management, access controls and general development principles.</p>	
Scripts	<p>Project administrators can write scripts to manage the way users interact with a model, such as managing security, staff compliance and model access, and monitoring changes made by users.</p> <p>Administrators can also use workflow scripts to control a user's capacity to change a model element, taking into account factors such as access rights, group membership and even the value of a proposed change.</p>	Workflow Script Functions ^[232]

Topic	Detail	See also
Script Implementation	<p>When a model is launched, the Workflow Engine is initialized with the current user and group memberships. This information determines who can access and modify parts of a given model.</p> <p>When a selected event occurs, the script engine is initialized with values including the author's name and access rights, and the element name and version details.</p> <p>The workflow script implements rules governing change management, access control and model validation. If a user attempts to make changes in violation of company policy, the script denies the update.</p> <p>The user is notified why the validation failed and the activity is logged. These reminders help to reinforce company policy, reduce human error and provide management with valuable project feedback.</p>	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Admin Workflow** permission to develop and manage workflow scripts

Learn More:

- [List of Available Permissions](#) ^[205]

3.4.4.1 Workflow Script Functions

Workflow scripts are executed by the Enterprise Architect workflow engine, to manage user input. You write the scripts in the Scripting window, in VBScript, under the Workflow group type.

Topics:

Topic	Detail	See also
Functions for User Input	<p>Functions that Enterprise Architect calls to validate and control user input</p> <p>For each of the functions that Enterprise Architect calls, a set of objects are filled</p>	Functions - Validate and Control User Input ^[233] Filled Workflow Data Structures ^[234]
Functions to create a Search	Functions that Enterprise Architect calls to create a search with user tasks	Functions - Create a Search With User Tasks ^[234]
Workflow Data Structures Enterprise Architect fills	Workflow data structure objects that Enterprise Architect fills	Filled Workflow Data Structures ^[234]
Workflow Data Structures you fill	Workflow data structure objects that you can fill	Workflow Data Structures You Fill ^[235]

Topic	Detail	See also
Functions you call	Functions that Enterprise Architect provides for you to call	Functions You Call [236]

Notes:

- If you make changes to a workflow script listed in the Scripting window, click on the **Refresh Scripts** button in the Scripting window toolbar to reload the script with the changes

Learn More:

- [Scripting](#) [1817]
- [Script Group Properties](#) [1819]

3.4.4.1.1 Functions - Validate and Control User Input

These are functions that Enterprise Architect calls to validate and control user input. For each function a set of objects is filled.

Function	Action	Return Value
AllowPhaseUpdate(OldValue, NewValue)	Validate a change a user has made to a phase	<ul style="list-style-type: none"> True to allow this user to make this change False to disallow the change and revert to the previous value
AllowStatusUpdate(OldValue, NewValue)	Validate a change a user has made to a status	
AllowTagUpdate(TagName, OldValue, NewValue)	Validate a change a user has made to a Tagged Value	
AllowVersionUpdate(OldValue, NewValue)	Validate a change a user has made to a version	
CanEditPhase()	Enable or disable the control for editing a phase	<ul style="list-style-type: none"> True to allow this user to make changes by enabling the control False to completely disable edit of this property by disabling the control
CanEditStatus()	Enable or disable the control for editing a status	
CanEditTag(TagName)	Enable or disable the control for editing a Tagged Value	
CanEditVersion()	Enable or disable the control for editing a version	
PreAllowPhaseUpdate(OldValue, NewValue)	Determine what information is required to validate this change	Semi-colon separated list of additional data required in order to validate this change <i>Supported Data Type:</i> Tests - fill the Tests array in the WorkflowContext object
PreAllowStatusUpdate(OldValue, NewValue)		
PreAllowTagUpdate(TagName, OldValue, NewValue)		
PreAllowVersionUpdate(OldValue, NewValue)		

Learn More:

- [Filled Workflow Data Structures](#) ^[234]

3.4.4.1.2 Functions - Create a Search With User Tasks

These are functions that Enterprise Architect calls to create a search with user tasks.

Function	Action	Return Value
GetWorkflowTasks	Describe the searches that this user must run	Ignored

3.4.4.1.3 Filled Workflow Data Structures

These workflow data structures are objects that Enterprise Architect fills.

Workflow Data Structure	Description	Calls	See Also
WorkflowUser	<p>This object provides information about the user currently logged in to the model</p> <p>It is filled by Enterprise Architect before any function is called by Enterprise Architect; it has the following properties:</p> <ul style="list-style-type: none"> • Username - the username for login to the system (if using Windows Authentication, this matches the Windows username) • Firstname - as found in the Security Users dialog • Surname - as found in the Security Users dialog • Fullname - the combination <Firstname> <Surname> (the form Enterprise Architect uses for Author fields and similar) 	This object calls the IsMemberOf (GroupName) function (see below)	Maintain Users ^[198]
WorkflowContext	<p>This object provides information about the object currently in context</p> <p>It is filled by Enterprise Architect before any searches except GetWorkflowTasks are run; it has the following properties:</p> <ul style="list-style-type: none"> • MetaType - the type of the current object, either an Enterprise Architect core type or a profile-specified metatype • Name - as found in the object Properties dialog • Status - as found in the object Properties dialog • Phase - as found in the object Properties dialog • Version - as found in the object Properties 	This object calls the TagValue (TagName) function (see below)	Functions - Create a Search With User Tasks ^[234] The Testing Workspace ^[1695]

Workflow Data Structure	Description	Calls	See Also
	dialog <ul style="list-style-type: none"> • Stereotypes - an array of strings for the stereotypes applied to this object • Tags - an array of Tagged Values, providing: <ul style="list-style-type: none"> • Name - the Tagged Value name • Value - the Tagged Value value • Tests - an array of tests; only filled during an <i>Allow*</i> call after the <i>PreAllow*</i> call has specified that tests are required Provides the following details, as found in the Testing window: <ul style="list-style-type: none"> • Name • Status • RunBy • CheckedBy • TestClass • TestType 		

Function	Action	Return Value
IsMemberOf (GroupName)	Check group membership of the current user	True if the current user is a member of the group with the specified name
TagValue (TagName)	Get the value from a named tag	Returns the value of the first Tagged Value with that name, or an empty string if no Tagged Value with that name exists

3.4.4.1.4 Workflow Data Structures You Fill

These two workflow data structures are objects that you can fill.

Workflow Data Structure	Description
WorkflowStatus	Use this data structure to provide information on the status of the object <ul style="list-style-type: none"> • LogEntry - set to True or False to indicate whether or not a log item should be recorded • Reason - indicate what reason should be recorded in the log • Action - indicate how to display the log message; valid values are: MessageBox, StatusBar, Output (default)
WorkflowSearches	Use this data structure to provide an array of searches Use Redim WorkflowSearches(x) to specify the number of searches being provided Each search has the following attributes: <ul style="list-style-type: none"> • Name - the name of this search • Group - the name of the group that this search should appear under in the Search

Workflow Data Structure	Description
	<p>combo box</p> <ul style="list-style-type: none"> • ID - the GUID for this search • Tasks - the array of tasks that this search looks for; an entry describes how to find all objects required to meet a particular task: <ul style="list-style-type: none"> • Name - the name of the task, as displayed in the Search view; workflow searches are grouped by this field by default • Conditions - an array of conditions, all of which must be matched for an object to be included in this task; a condition is a comparison of a single field to a value: <ul style="list-style-type: none"> • Column - the name of the field • Operator - operator types, either = (provide matching values only) or <> (provide non-matching values only) • Value - if this contains a comma, the string is treated as a comma separated list of values to compare against; otherwise the string is a single value to compare against

3.4.4.1.5 Functions You Call

These are functions that Enterprise Architect provides for you to call.

Function	Action	Return Value	See also
NewSearch(name, group, guid, taskcount)	Create a new search object to be included in WorkflowSearches Initializes each member	The created search	Workflow Data Structures You Fill ^[235]
NewTask(name, conditioncount)	Create a new task object to be included in a search Initializes each member	The created task	
NewCondition(column, operator, value)	Create a new condition object to be included in a task Initializes each member	The created condition	
SetLastError(message, outputMethod)	Log and/or report the provided message to the user	The message	

3.4.5 Sharing Reference Data

You can conveniently update your models with reference data (including Glossary and Issue information) by exporting the data to and importing the data from .XML files.

You import data into the model automatically or manually from a reference data .XML file, exported from another model or an iteration of the current model.

Use to: (For example)

- Copy glossaries from one model to another
- Add additional stereotype profiles by merging new stereotypes into the model
- Update reference data from files supplied by Sparx Systems as a maintenance release
- Copy resources, clients and so on from one model to another

Learn More:

- [Reference Data](#)^[772]
- [Import Reference Data](#)^[239]
- [Export Reference Data](#)^[237]

3.4.5.1 Export Reference Data

This topic explains how to export reference and project data to a custom XML file. This includes table information, filter information, rows and columns.

Access: [Project | Model Import/Export | Export Reference Data](#)

Use to:

- Capture reference data of one or more specific data types in an XML file, to review or transfer into another project or tool
The exported data includes all instances of the data type in the project; for example, all defined cardinality values, or all RTF Style Templates

How to:

Step	Action	See also
1	Select the Export Reference Data menu option The Data Exporter dialog displays	
2	From the Name list, select the data type tables to export You can select one or more tables to be exported to a single XML file, by pressing (Ctrl) or (Shift) as you click on the data type table names	Code Templates ^[1482] Automation Scripts ^[1817] (JavaScript, JScript and VBScript) Cardinality Types ^[776] Constraint Status Types ^[785] CSV ^[338] Specifications Defined Metric Types ^[354] Defined Problem Types ^[787] Diagram Matrix Profiles ^[592] (Model Profiles) Estimation - Environment Complexity Factor Values ^[402] Estimation - Technical Complexity Factor Values ^[401] General Constraint Types ^[784] General Effort Types ^[353] Import Component Types ^[1518] Linked Document Templates ^[734] Macro List ^[1525] (Preprocessor)

Step	Action	See also
		<p>macros)</p> <p>Maintenance Types ^[787] (Problem Types plus Test Types)</p> <p>Model Authors ^[778]</p> <p>Model Data Types - Code ^[777] and DDL ^[1355]</p> <p>Model Images ^[593]</p> <p>Project Clients ^[782]</p> <p>Project Glossary ^[362]</p> <p>Project Issues ^[358]</p> <p>Project Resources ^[787]</p> <p>Project Roles ^[780]</p> <p>Project Tasks ^[356]</p> <p>Property Types ^[1108] (Tagged Value Types)</p> <p>Requirement Types ^[785]</p> <p>Risk Types ^[355]</p> <p>Scenario Types ^[786]</p> <p>Security - Group Permission ^[204]</p> <p>Security - Groups ^[203]</p> <p>Security User Groups ^[207]</p> <p>Security - User Permissions ^[207]</p> <p>Security - Users ^[198]</p> <p>Standard Complexity Types (a Predefined Reference Data Tagged Value Type ^[1112])</p> <p>Status Colors ^[783] (colors defined for status types)</p> <p>Status Types ^[783]</p> <p>Status Types - Application ^[783] (elements to which status can be applied)</p> <p>Stereotypes ^[1039] (as listed on the Stereotypes page of the UML Types dialog)</p> <p>Templates - HTML Style (the web templates ^[1804] listed in the Resources window)</p> <p>Templates - HTML Style Detail (the content of the HTML report templates)</p> <p>Templates - RTF Document (the Extended RTF Style templates ^[1736] in</p>

Step	Action	See also
		<p>the Resources window)</p> <p>Templates - RTF Style (the Legacy RTF Style templates^[1792] in the Resources window)</p> <p>Templates - RTF Style Detail (the content of the Legacy RTF Style templates)</p> <p>Templates - RTF Tag and Language Options (RTF word substitution^[1794] templates)</p> <p>Test Types^[788]</p> <p>Transform Templates^[1330]</p>
3	Click on the Export button	
4	When prompted to do so, enter a valid file name with a .XML extension	
5	Click on the Save and OK buttons This exports the data to the file; you can use any text or XML viewer to examine the file	

Notes:

- If there are no instances of a selected data type in the project, the export does not generate any output for that data type in the XML file
- Entries for code language templates and/or transformation templates are included in the **Name** list only if templates for a particular language or transformation exist in the model
- Currently, Standard Complexity Types cannot be directly edited and are therefore effectively standard for all models; they can be listed using the Predefined Reference Data Tagged Value type *ComplexityTypes*

3.4.5.2 Import Reference Data

You can import reference data into a model in Enterprise Architect from a reference data .xml file that was exported from another model or from an iteration of the current model, either:

- Manually, as required, whenever you know there is new or changed data to apply, or
- Automatically whenever the model is reloaded into Enterprise Architect (if the file has changed since the previous import)

When you import data into Enterprise Architect, the system merges the incoming data with the existing data. If a record already exists it is updated to the new values. If the record does not exist, Enterprise Architect adds a new record. Enterprise Architect never deletes records.

The automatic import checks if the source file has changed since the last import; if the file has not changed, the import does not proceed. If the file has changed, the changed data is imported; however, you can configure Enterprise Architect to display a prompt for you to allow or cancel the import.

Access: **Project | Model Import/Export | Import Reference Data > Import File tab**
Project | Model Import/Export | Import Reference Data > Shared File tab

How to:

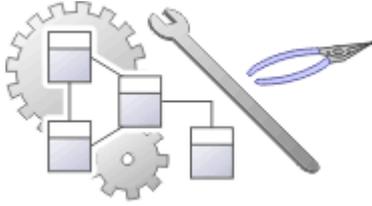
To import reference data *manually*, follow the steps below:

Step	Action	See also
1	Select the Import Reference Data menu option The Import Reference Data dialog displays	
2	Click on the Import File tab and on the Select File button, then select the filename to import data from This must be an XML file produced by the Enterprise Architect Data Exporter	Export Reference Data ^[237]
3	If you have entered the name of a valid file, a list of available tables to import displays in the Select Datasets to Import panel	Export Reference Data ^[237]
4	Click on one or more of the tables to import Press (Ctrl) or (Shift) to click on multiple tables	
5	Click on the Import button to start the process A message displays when the import is complete; generally the process is quite fast.	

To import reference data automatically, follow the steps below:

Step	Action	See also
1	Select the Import Reference Data menu option The Import Reference Data dialog displays	
2	Click on the Shared File tab	
3	If you are changing an existing configuration to import from a different XML file, click on the Clear button to clear the dialog fields	
4	Click on the Select File button and browse for the filename to import data from This must be an XML file produced by the Enterprise Architect Data Exporter	Export Reference Data ^[237]
5	If you have entered the name of a valid file, a list of tables to import displays in the Datasets in File panel	Export Reference Data ^[237]
6	If you prefer to control whether or not the automatic import takes place, select the Always prompt before import checkbox	
7	Click on the Import button to import the reference data now, and to enable the automatic check and import for subsequent reloads	

3.5 Change Management



This section describes the tools and facilities for controlling and monitoring changes to the data in a project.

Topics:

Topic	Detail	See also
Version Control of Packages	<p>Enterprise Architect Model Version Control enables you to:</p> <ul style="list-style-type: none"> • Coordinate sharing of packages between users, with either read-only access or update access, ensuring that work on different areas of the model is coordinated and synchronous rather than conflicting • Save and retrieve a history of changes to packages <p>To use version control in Enterprise Architect, you require a third-party source-code control application such as:</p> <ul style="list-style-type: none"> • Subversion • CVS • MS Team Foundation Server (TFS), or • Any other version control product that complies with the Microsoft Common Source Code Control standard 	Version Control ^[242]
Tracking Changes	<p>Enterprise Architect provides two separate but complementary facilities for tracking changes to data across the project:</p> <ul style="list-style-type: none"> • Auditing of model changes • Baselining and differencing to capture and roll back changes 	Tracking Changes ^[298]
Project Data Transfer	<p>Enterprise Architect enables you to transfer project data between project data repositories either for:</p> <ul style="list-style-type: none"> • sections of the project (XML and CSV) or • the whole project, row by row, table by table (in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect) 	Model Transfer ^[318]

Learn More:

- [Check Project Data Integrity](#)^[414]
- [Maintenance](#)^[414]
- [Team Development](#)^[185]
- [Spell Checking](#)^[372]
- [Reference Data](#)^[772]
- [Project Upgrade](#)^[417]

3.5.1 Version Control

Enterprise Architect supports version control of your projects, by allowing you to place your model packages under version control. You can place any individual packages, view nodes or model root nodes under version control.

To use version control in Enterprise Architect, a third-party source-code control application is required that controls access to and stores revisions of the controlled packages.

Guide:

Topic	Detail	See Also
Introduction	Version Control provides numerous key facilities: <ul style="list-style-type: none"> • Saving a history of changes to Enterprise Architect packages • The ability to retrieve previous revisions of packages • Propagating model updates between team members • Coordinating sharing of packages between team members 	Introduction ^[243]
Product Setup	To use version control in Enterprise Architect, a third-party source-code control application is required that controls access to and stores revisions of the controlled packages	Setting Up a Version Control Environment ^[252]
Configuring Enterprise Architect	Once the version control software has been installed and configured, to use your installed version control product you must first define a Version Control Configuration within Enterprise Architect	Version Control Setup ^[268]
Using Version Control	Version control activities include such things as; <ul style="list-style-type: none"> • Controlling packages • Checking out packages for editing • Checking in modifications to packages • Retrieving previous revisions of controlled packages 	Using Version Control ^[275]

Notes:

- Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect; it is possible to leave the package files in a state that Enterprise Architect cannot recognize
- Database replication should not be combined with version controlled packages
- If the packages under version control contain any alternative images and those images are subject to frequent change, you can set the **Export alternate images** option on the Options dialog to export the images to the version control repository when you check in the packages; if the images are not subject to frequent change, do not select this option and instead use **Export / Import Reference Data** to manage alternative images

3.5.1.1 Introduction

Enterprise Architect's version control integration provides numerous key facilities;

- Saving a history of changes to your model's packages
- The ability to retrieve previous revisions of packages
- Propagating model updates between team members
- Coordinating sharing of packages between team members

This section discusses some factors you should consider when setting up and using version control in your model development.

Topics:

Topic	Detail	See also
Version Control Usage	<p>There are two main ways in which Enterprise Architect's projects can be deployed;</p> <ul style="list-style-type: none"> • Centralized Shared Model • Distributed Private Models <p>Version control is employed in the same way for both scenarios, however, when using Private Model deployment it serves the additional role of propagating model updates throughout the team.</p> <p>Version Control can also be used to share standard packages between different projects.</p>	Version Control Usage [244]
Version Control Basics	Enterprise Architect must enforce serialized editing of version controlled packages, using the lock-modify-unlock mode of operation.	Version Control Basics [246]
Applying Version Control to Models	Using version control in Enterprise Architect consists of placing individual model packages under version control, rather than version controlling the project as a whole.	Version Control Of Model Data [245]
Version Control and Project Reference Data	To ensure that changes in reference data are shared between users in a version-controlled project that is deployed as multiple private models, you should periodically export the reference data from the model where the changes were made, and import it into the other models maintained by the team.	Version Control and Reference Data [246]
Team Deployment		Version Control & Team Deployment [247]
Project Browser Indicators	Packages under version control are identified in the Project Browser by icons that indicate the current status of the package.	Project Browser Indicators [249]
Offline Version Control	<p>You can prevent Enterprise Architect from attempting to make any version control connections by choosing to Work Offline before loading a model.</p> <p>If Enterprise Architect is unable to connect a Version Control Configuration for any reason, it displays warning messages to notify you and provides 'offline' version control functionality for all packages associated with the failed connection.</p>	

Topic	Detail	See also
System Requirements and Configuration	<p>To use version control in Enterprise Architect, a third-party source-code control application is required that controls access to and stores revisions of the controlled packages.</p> <p>Typically there are:</p> <ul style="list-style-type: none"> • A server component that manages a version control repository, and • Client components on the workstations, that manage local working copies of controlled files. <p>Enterprise Architect uses the client component to communicate with the server. A version control client must be installed on every machine where you run Enterprise Architect and want to access your version control system.</p>	System Requirements and Configuration ^[25]

3.5.1.1.1 Version Control Usage

There are four basic ways in which the version control facility might be used:

Field	Usage	See also
Single Shared model	<p>Users share an Enterprise Architect model, stored in a central .EAP file or DBMS repository. This configuration enables you to view changes to other users' packages without explicitly having to check them out, but by simply refreshing your view of the model.</p> <p>Version control is used to;</p> <ul style="list-style-type: none"> • Archive successive versions of your work to date • Maintain package revision history • Provide an "undo" facility to recover from unwanted changes or accidental deletions • Regulate access to packages 	
Multiple Private models	<p>An Enterprise Architect model is created by a single user who configures it for version control. The model file is then distributed to other users, with each user storing their own private copy of the model.</p> <p>Version control is used to;</p> <ul style="list-style-type: none"> • Propagate changes to the model, throughout the team • Archive successive versions of your work to date • Maintain package revision history • Provide an 'undo' facility to recover from unwanted changes or accidental deletions • Regulate access to packages 	
Shared packages	<p>Individual users create separate Enterprise Architect models but share one or more packages.</p> <ul style="list-style-type: none"> • Users share packages through version control 	
Standard packages	<p>A company might have a standard set of packages which are broadly shared (on a read-only basis).</p> <ul style="list-style-type: none"> • Individual users retrieve packages with the Get Package menu option 	

Learn More:

- [Version Control Best Practices for Enterprise Architect](#)

3.5.1.1.2 Version Control of Model Data

Using version control in Enterprise Architect consists of placing individual model packages under version control, rather than version controlling the project as a whole.

Guide:

Topic	Detail	See also
Version Controlling your Project files	<p>All Enterprise Architect models are stored in databases - even the .EAP file is an MS Jet database</p> <p>In simple terms, the project file is a single entity of binary data</p> <p>It is not practical to apply version control to the database as a whole; being binary data, it would require the use of the lock-modify-unlock model of version control, which would mean that only a single user at a time could work on any given (version controlled) model</p> <p>Therefore, avoid placing your .EAP files under version control, as this can create problems for you</p> <ul style="list-style-type: none"> • Most version control systems mark their controlled files as read only, unless they are specifically checked-out to you • The .EAP file is an MS Jet database, and Enterprise Architect must be able to open this file for read/write access when you load your model (Enterprise Architect displays an error message and fails to load the model if it is read-only) 	Version Control Basics ^[246]
Version Controlling Packages in your model	<p>To overcome the limitations described above, Enterprise Architect exports discrete units of the model - the packages - as XML package files, and it is these XML files, not the .EAP file, that are placed under version control</p> <p>The XML file format used by Enterprise Architect dictates that they too be treated as binary files (therefore it is not possible to merge the XML files either); however, by splitting the model into much smaller parts, this approach enables many users to work on separate parts of the model simultaneously</p>	
Nested version controlled packages	<p>Nested version controlled packages result in much smaller XML files being exported for parent packages, as the parent packages' XML files do not contain any content for the version controlled child packages</p> <p>Version Control of nested packages, together with a model structure having small individual packages, also provides greater scope for multiple users to work concurrently, as individual users are locking much smaller parts of the model</p>	

3.5.1.1.3 Version Control and Reference data

To ensure that changes in reference data are shared between users in a version-controlled project that is deployed as multiple private models, you should periodically export the reference data from the model where the changes were made, and import it into the other models maintained by the team.

Guide:

Topic	Detail	See also
Reference Data	Reference data is data that is used across a model or project; it is not package-specific.	Reference data ^[772]
Version Control in Enterprise Architect	Version control in Enterprise Architect operates at package level, and therefore does not capture changes in reference data. Where version control is used in a multiple private model set up, changes in reference data are not brought into the model when packages are updated from version control.	
Sharing Reference Data	In a Shared Model environment, all users are accessing the same Project Reference data. However, in a Multiple Private Model environment, to ensure that changes in reference data are shared between users in a version-controlled project, you should periodically export the reference data from the model where the changes were made, and import it into the other models maintained by the team.	Sharing Reference Data ^[236]
Version control of Reference Data	Reference data is exported and imported as an XML file, which contains whatever types of reference data you want to transfer. To place your project reference data under version control, you should export the data as an XML file and apply version control to the file using your version control software, external to Enterprise Architect	Export Reference Data ^[237]

3.5.1.1.4 Version Control Basics

Enterprise Architect implements version control of your model, by exporting package data from the project database to XML package files. It is these package files that are then placed under version control. The XML file format cannot be merged in the same way that ordinary text files can be merged.

The information below, illustrates why Enterprise Architect must enforce serialized editing of version controlled packages.

Guide:

Topic	Detail	See also
The Lock-Modify-Unlock Solution	Many version control systems use a lock-modify-unlock model to address the problem of different authors in a shared source overwriting each other's work. In this model, the version control repository allows only one person to change a file at a time, and access is managed using locks. Harry must lock a file before he can begin making changes to it. If Harry has locked a file, Sally cannot also lock it, and therefore cannot make any changes to that file. All she can do is read the file, and wait for Harry to finish his changes and release the lock. After Harry unlocks the file, Sally can take her turn in	

Topic	Detail	See also
	locking and editing the file.	
The Copy-Modify-Merge Solution	<p>Subversion, CVS and a number of other version control systems use a copy-modify-merge model as an alternative to locking.</p> <p>In this model, each user's client contacts the project repository and creates a personal working copy- a local reflection of the repository's files and directories. Users then work simultaneously and independently, modifying their private copies. In due course, the private copies are merged together into a new, final version. The version control system often assists with the merging, but ultimately a person is responsible for making it happen correctly.</p>	
When Locking is Necessary	<p>While the lock-modify-unlock model is generally considered a hindrance to collaboration, there are still times when locking is necessary.</p> <p>The copy-modify-merge model is based on the assumption that files are contextually merge-able: that is, the files in the repository are line-based text files (such as program source code). But for files with binary formats, such as artwork or sound, it is often impossible to merge conflicting changes. In these situations, it really is necessary for users to take strict turns in changing the file. Without serialized access, somebody ends up wasting time on changes that are ultimately discarded.</p>	

3.5.1.1.5 Applying Version Control in a Team Environment

The following information provides an overview of the steps involved in setting up a version control environment and applying version control to an Enterprise Architect project to be accessed by a number of users.

Use to:

- Setup a version control environment
- Apply version control to an Enterprise Architect project

How to:

Step	Action	See also
1	Install your version control product.	
2	Create a version control repository.	
3	Create a version control project to be used with your Enterprise Architect project	
4	<p>Check-out a working copy of the version control project (a module, project or folder within the version control system) into a local folder.</p> <p>(You must do this for every team member that is accessing the version controlled packages, whether you are using a single shared model or each team member stores his own private copy of the model.)</p>	Create a Local Workspace - SVN ^[256] Create a Local Workspace - CVS ^[262]
5	Within Enterprise Architect, define a version control configuration to provide access to	Version

	<p>the working copy files.</p> <p>Throughout a team, the name of the version control configuration must be the same across all machines. That is, all version control access to a given Enterprise Architect package must be through version control configurations with the same name, across all models and all users.</p> <p>The easiest way to perform this step, (throughout the team), is to have one user set up version control on the model and then share that model with the rest of the team.</p> <ul style="list-style-type: none"> • In Shared Model deployment, all users connect to a single instance of the model database, so the model is shared automatically. • In Private Model deployment, it is easiest to distribute copies of the original model (after version control has been set up) to all other members of the team. <p>Whenever you open a model (Private or Shared) that uses a version control configuration that is not yet defined on your workstation, Enterprise Architect prompts you to complete the definition for that configuration. This typically means specifying the local working copy directory and maybe choosing the version control project associated with this Enterprise Architect project.</p> <p>Once this has been done, the version controlled packages that already exist in the model are ready for use.</p>	Control Setup ^[268]
6	Configure packages within the Enterprise Architect model for version control. That is, apply version control to individual packages.	Configure Package for Version Control ^[282]
7	Check-out and check-in packages as required.	Check Out ^[285] Check In ^[285]

Notes:

- It is possible to use multiple version control configurations within the same model, so different packages can still use different version control configurations within the same model, as long as any given package is always accessed via the same version control configuration

Learn More:

Team deployment and the use of version control is discussed in two Sparx Systems white papers, available on the Sparx Systems web site:

- http://www.sparxsystems.com/WhitePapers/Version_Control.pdf
- http://www.sparxsystems.com/downloads/whitepapers/EA_Deployment.pdf

3.5.1.1.6 Version Control Nested Packages

In releases of Enterprise Architect later than version 4.5, when you save a package to the version control system only stub information is exported for any nested packages. This ensures that information in a nested package is not inadvertently over-written by a top level package.

Guide:

Topic	Detail	See also
Checking Out	When checking out a package, Enterprise Architect does not modify or delete nested packages; only the top level package is	

Topic	Detail	See also
	<p>modified.</p> <p>As a consequence of this behavior, if you check out or get a version controlled package with nested packages not already in your model, you see stubs in the model for the nested packages only.</p>	
Get All Latest	If you select the Get All Latest option from the version control menu, Enterprise Architect populates any new stubs from the version control system.	Get All Latest ^[278]
Importing Models	<p>You can populate a large and complex model, by 'getting' only the root packages, then using Get All Latest to recursively iterate through the attached and nested packages.</p> <p>This is a powerful and efficient means of managing your project and simplifies handling very large models, even in a distributed environment.</p> <p>The command Import a Model Branch, combines the steps described above into a single operation.</p>	Get Package ^[289] Import Controlled Model Branch ^[297]

Notes:

- It is recommended you do not mix versions of Enterprise Architect later than version 4.5 with earlier versions when sharing a version controlled model. If this is necessary it is best to go to the Version Control Settings dialog and deselect the Save nested version controlled packages to stubs only checkbox, setting Enterprise Architect to the pre-version 4.5 behavior (for the current model only)

3.5.1.1.7 Project Browser Indicators

Packages under version control are identified in the Project Browser by icons that indicate the current status of the package.

Reference:

Icon	Indicates that	See Also
	This package is version controlled and not checked out to you. You cannot edit the package (unless you check out the package).	
	This package is version controlled and checked out to you. You can edit the package.	
	This package is version controlled, but you checked it out whilst not connected to the version control server. You can edit the package but there could be version conflicts when you check the package in again.	Offline Version Control ^[250]
	This package is controlled and is represented by an XMI file on disk, but it is not under version control. You can edit this package.	Controlled Packages ^[327]

Learn More:

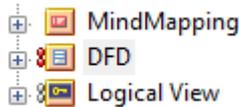
- [Using Version Control](#) ^[275]

3.5.1.1.8 Offline Version Control

When loading a model that uses version control, Enterprise Architect normally initializes a connection to the version control system for each Version Control Configuration defined in the model.

If Enterprise Architect is unable to connect a Version Control Configuration for any reason, it displays warning messages to notify you and provides 'offline' version control functionality for all packages associated with the failed connection.

Guide:

Topic	Detail	See also
Choose to Work Offline	<p>You can prevent Enterprise Architect from attempting to make any version control connections by selecting the Project Version Control Work Offline menu option before loading a model.</p> <p>This is useful if you know that Enterprise Architect cannot connect to your version control system. For example, if you are working on a laptop computer that is disconnected from your network and you have an Enterprise Architect model that uses a large number of Version Control Configurations, choosing to work offline before you load the model enables you to avoid all the error messages that Enterprise Architect would normally display as each version control connection attempt fails.</p> <p>You can switch between working offline and working online at any time, either before or after a model is loaded. Toggle the Project Version Control Work Offline menu option. Enterprise Architect disconnects or reconnects version control (depending on connection availability) according to your selection.</p>	
Use Version Control Whilst Disconnected From Your Version Control Server	<p>Enterprise Architect 'remembers' the status of a model's version controlled packages. Packages that were checked out to you prior to disconnecting from the server are still shown as checked out to you, even though you are no longer connected to the server. You can still edit these packages as you normally would.</p> <p>Packages that were not checked out to you prior to disconnecting from the server are shown as version controlled and locked. You cannot edit these packages until you check them out.</p>	
Offline Check Out	<p>In releases of Enterprise Architect from release 6.0 onwards, you can 'check-out' and edit a version controlled package even when your machine is disconnected from the version control server. In the example below, the colored 'figure 8' icon for DFD indicates that you have checked it out whilst offline (the gray 'figure 8' icon shown against Logical View indicates that you have checked out a version-controlled package online).</p> 	Project Browser Indicators ^[249]

Topic	Detail	See also
	<p>You should be aware that the version control system - and therefore other users - have no way of knowing that you have 'checked-out' a package whilst offline. It is not possible to merge changes to an XML file that result from two users editing the same package at the same time. If an offline checkout leads to two people editing the same package at the same time, when the changes are brought back online the first-saved set of changes is lost.</p>	
<p>Checking In a Package That Was Checked Out Offline</p>	<p>Once you reconnect your machine to the version control server, if the package you checked out offline is not currently checked out by another user, you can check in that package. However, before Enterprise Architect checks in such a package, it compares the local working copy of the package file with the latest revision in the repository. (These package files remain unchanged in your work area until Enterprise Architect exports the package again before checking in.) If the repository version remains unchanged from when you last updated your local copy, Enterprise Architect exports and checks in your package without further prompting.</p> <p>On the other hand, if the repository now contains a file that has changed since you last updated your local copy, checking in your package overwrites whatever those changes might be. Enterprise Architect displays a message warning you of the pending data loss and giving you the opportunity to abort the check in. At this point, you must decide whether to discard your own changes, using the Undo Check Out command, or continue with your check in and overwrite the changes that have been committed to the repository since you last updated your local copy from the repository.</p> <p>You can use the File Properties command to determine who checked in the last changes to this package. This might help you to discover what changes have been uploaded and decide whose changes take precedence.</p>	
<p>Update Before You Disconnect</p>	<p>Whenever you are connected to the version control server, you are always working with the latest version of a package. This is because you cannot modify a package until you check it out from version control, and checking it out loads the latest revision from the repository into your model.</p> <p>These rules do not apply when you are disconnected from the version control server. You are working on whatever versions you have on your machine, dating back to the last time you updated your local copy of each version controlled package. So, if you are planning to work on a model whilst disconnected from version control, it is a very good idea to make sure that you have the latest versions of all packages before you disconnect. The Get All Latest option makes this a simple task.</p>	<p>Get All Latest^[278]</p>

3.5.1.1.9 Version Control Branching

Currently, Enterprise Architect does not support Version Control Branching.

Work-arounds to achieve similar results might be possible for certain version-control products; contact Sparx Support for advice:

Topic	Link
Registered users	http://www.sparxsystems.com/registered/reg_support.html
Trial users	support@sparxsystems.com

3.5.1.2 Version Control Product Setup

Enterprise Architect relies on third-party version control products to control and maintain the different revisions of your project packages.

Once your version control product is installed and a suitable environment has been created, Enterprise Architect can then utilize that environment to control your project's packages.

Guide:

Topic	Detail	See also
Version Control System Components	<p>Typically, version control products consist of:</p> <ul style="list-style-type: none"> • a server component • a client component <p>Enterprise Architect integrates with version control client components.</p> <p>It provides integration for Subversion, CVS and MS Team Foundation Server command line clients, as well as integration for products having API clients that comply with the MS SCCI specification.</p>	
Version Control Server	<p>The server component maintains the controlled files in their many revisions in a central repository.</p> <p>The server component is usually located on a server machine, that is accessible by all team members who are using version control.</p>	
Server Configuration	<p>Steps involved in configuring a version control server usually include;</p> <ul style="list-style-type: none"> • Installing the software • Creating a repository • Creating version control projects (or modules or folders for use with specific projects) • Configuring user IDs and passwords <p>The details of configuring any particular version control server are beyond the scope of this document.</p> <p>Please consult the appropriate documentation.</p>	
Version Control Client	<p>The client component deals with the working copies of the controlled files, submitting or retrieving files to and from the server as required.</p> <p>A version control client must be installed on every machine where you run Enterprise Architect and want to access your version control system.</p>	
Client Configuration	<p>Steps involved in configuring a version control client usually include;</p> <ul style="list-style-type: none"> • Installing the software • Creating a new directory for use as a local working copy 	<p>Create a Subversion Environment^[254]</p> <p>Create a CVS</p>

Topic	Detail	See also
	<p>folder</p> <ul style="list-style-type: none"> Logging in to the version control server Associating the working copy folder with its corresponding server repository folder <p>For specific information on how to set up a version control environment for use with Enterprise Architect, follow the appropriate link at right.</p>	<p>Environment^[260]</p> <p>Create a TFS Environment^[264]</p> <p>Create an SCC Environment^[267]</p>

3.5.1.2.1 System Requirements

Enterprise Architect is a Windows based application and requires a Windows based version control client for integration.

Enterprise Architect is independent of the version control server component and the platform on which that runs.

Reference:

Product	Detail	See Also
Subversion	<p>Subversion is free, open source software</p> <p>Subversion server components are available to run on a wide range of different hardware and operating systems; Enterprise Architect is not affected by your choice of server components</p> <p>Enterprise Architect requires Subversion's Windows-based command line client for integration</p> <p>There are numerous graphical user interface clients available for use with Subversion, such as TortoiseSVN; this type of client cannot be used for integration with Enterprise Architect, but they can be very useful in preparing a working Subversion environment for use by Enterprise Architect</p> <p>Binary packages are available for download from:</p> <ul style="list-style-type: none"> http://subversion.apache.org/packages.html <p>Subversion documentation is available from:</p> <ul style="list-style-type: none"> http://svnbook.red-bean.com/nightly/en/index.html 	<p>Create a Subversion Environment^[254]</p>
CVS	<p>CVS is free, open source software</p> <p>CVS server components are available to run on a wide range of different hardware and operating systems; Enterprise Architect is not affected by your choice of server components</p> <p>Enterprise Architect requires CVS's Windows-based command line client for integration</p> <p>There are numerous graphical user interface clients available for use with CVS, such as TortoiseCVS; this type of client cannot be used for integration with Enterprise Architect, but they can be very useful in preparing a working CVS environment for use by</p>	<p>Create a CVS Environment^[260]</p>

Product	Detail	See Also
	<p>Enterprise Architect</p> <p>The software is available for download from:</p> <ul style="list-style-type: none"> • http://www.nongnu.org/cvs/ <p>CVS documentation is available from:</p> <ul style="list-style-type: none"> • http://cvsbook.red-bean.com/cvsbook.html 	
Microsoft Team Foundation Server	<p>Enterprise Architect is able to use either the command line client for TFS, or the MS TFS-SCC client; your choice of client affects how you specify your Version Control Configuration</p> <p>MS TFS-SCC clients are available for download from Microsoft's web pages:</p> <ul style="list-style-type: none"> • Visual Studio 2005 Team Foundation Server MSSCCI Provider • Visual Studio Team System 2008 Team Foundation Server MSSCCI Provider 	<p>TFS Settings ^[274]</p> <p>SCC Settings ^[274]</p>
SCC compatible products	<p>Any version control product that provides a client that complies with the Microsoft Common Source Code Control standard, version 1.1 or higher, can be integrated with Enterprise Architect</p> <p>The following products are SCC-compatible and are known to successfully integrate with Enterprise Architect:</p> <ul style="list-style-type: none"> • Accurev • ClearCase • MS Visual Source Safe • MS TFS-SCC • MKS Source Integrity • Perforce • Source Offsite • Snapshot CM <p>Products that do not appear in the list should still integrate successfully with Enterprise Architect, if there is a client available for that product that complies with the MS SCC API specification</p>	

3.5.1.2.2 Create a Subversion Environment

Before Subversion can be used as a version control provider for Enterprise Architect, the appropriate software must be installed and configured by a Subversion administrator.

The following topics describe some basic tasks that must be completed and some tools for performing these tasks, in order to create an operational Subversion environment.

Topics:

Topic	Detail	See Also
Install server components	<p>Executable files for Subversion can be obtained from the Apache Software Foundation</p> <p>Subversion server components are available to run on a</p>	<p>Apache Subversion Official Subversion Documentation</p>

Topic	Detail	See Also
	<p>wide range of different hardware and operating systems; Enterprise Architect is not affected by your choice of server components</p> <p>VisualSVN is a package that can greatly simplify the installation, configuration and management of your Subversion server</p>	
Create a repository	Please consult the official Subversion documentation	Official Subversion Documentation
Create Subversion users	Please consult the official Subversion documentation	Official Subversion Documentation
Create a new repository sub-tree	<p>It is good practice to create folders in your PC's file system for organizing your files and documents; similarly, it is good practice to create a new repository sub-tree in Subversion for each new Enterprise Architect model</p> <p>TortoiseSVN can greatly simplify the process of creating new repository sub-trees</p>	Create a Repository Sub-tree ^[256]
Install client components	Executable files for Subversion can be obtained from the Apache Software Foundation	Apache Subversion
Create a working copy folder	<p>To create a local working copy, you must perform an initial check-out of a folder from the Subversion repository; this downloads a copy of the folder and its contents, to create your local working copy</p> <p>TortoiseSVN can greatly simplify the initial check out of a working copy folder</p>	Create a Local Working Copy ^[256]
Subversion under Wine/CrossOver	The set up and use of Subversion with Enterprise Architect under Wine is almost identical to when running natively under Windows, apart from some minor differences when installing the Subversion client and performing the initial check out of the working copy folder	Subversion under Wine ^[256]
TortoiseSVN	TortoiseSVN is a Windows shell extension; Enterprise Architect can not use TortoiseSVN as its client, it must use the Subversion command line client	TortoiseSVN ^[260]

Notes:

- Enterprise Architect relies on exclusive file locking when applying version control to its packages; file locking was not introduced into Subversion until version 1.2, therefore Enterprise Architect does not work with Subversion releases earlier than Subversion 1.2
- Sparx Systems recommend that each new Enterprise Architect model being added to version control with Subversion should have a separate repository sub-tree created for it, and users should create a new local working copy from the sub-tree to be used with that model

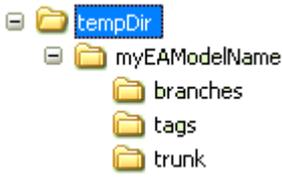
3.5.1.2.2.1 Create a new Repository Sub-tree

In the same way that it is good practice to create folders in your PC's file system for organizing your files and documents, it is good practice to create a new repository sub-tree in Subversion for each new Enterprise Architect model.

Use to:

- Create a new sub-tree in the Subversion repository

How to:

Step	Action	See also
1	<p>Use Windows Explorer to create a temporary directory structure on your PC file system, that can be imported into the Subversion repository to create a new repository sub-tree. The resulting sub-tree can then be used to control the package files for your Enterprise Architect model.</p> <p>The directory structure should look like this:</p> 	
2	<p>Open a Windows command prompt, navigate to "tempDir" and issue the Subversion command import.</p> <p>For example;</p> <pre>C:\Documents and Settings\user> cd \tempDir C:\tempDir> svn import . https://host.example.com:8443/repos/</pre>	Subversion repository administration basics
3	Delete the temporary directory structure. That is, delete "tempDir" and all its contents.	

Notes:

- After the import is finished, the original tree is not converted into a Subversion working copy; you should delete the temporary structure and check out a fresh working copy of the tree
- The process described above can also be performed using TortoiseSVN's Repository Browser, which provides commands that allow you to simply create new folders directly in the repository

3.5.1.2.2.2 Create a Local Working Copy

In order to make use of Subversion to provide version control of your model's packages, you must prepare a functional SVN working copy folder, that can be accessed through an Enterprise Architect version control configuration.

Use to:

- Create a Subversion working copy folder, suitable for use with Enterprise Architect.

How to:

Step	Action	See also
1	Choose a suitable directory on your system, in which to create your Subversion Working Copy.	
2	Open a command prompt window and (if necessary create and then) navigate to a suitable directory to use as your Subversion working copy directory. For example: <pre>C:\> mkdir mySVNWorkSpace C:\> mkdir mySVNWorkSpace/myEAModelName C:\> cd mySVNWorkSpace/myEAModelName</pre>	
3	Perform the initial check out of the Subversion repository folder, into the local working copy folder. For example; <pre>C:\mySVNWorkSpace> svn checkout https://host.example.com:8443/re</pre> A subdirectory is created in your current working directory, that has the same name as the folder being checked out. Any files already existing in the repository folder are downloaded to the working copy folder as working copy files.	
4	You should now verify that your Subversion environment functions correctly.	Verify the SVN Workspace 257

Notes:

- The process described above can also be performed using TortoiseSVN's Checkout command, which provides options to browse, when specifying both your repository folder and your local folder

3.5.1.2.2.3 Verify the SVN Workspace

After creating the Subversion local working copy to hold the working copies of your Enterprise Architect package files, you should verify that it functions correctly.

You should be able to add files to Subversion, lock the files and commit changes to those files.

Use To:

- Verify correct operation of a newly created Subversion working copy folder, before attempting to use it with Enterprise Architect.

How To:

Step	Action	See also
1	Use Windows to open a command prompt window.	
2	Change directory to the one you specified as the working copy, in the Subversion checkout command when preparing the SVN workspace. For example; <pre>C:\> cd mySVNWorkSpace</pre>	

3	<p>Create a test file, such as Test.txt, containing the text "Subversion Test".</p> <p>For example;</p> <pre>C:\> echo Subversion Test > Test.txt</pre>	
4	<p>Execute the following Subversion commands:</p> <ul style="list-style-type: none"> • svn add Test.txt • svn commit -m"Commit comment" Test.txt • svn update Test.txt • svn lock Test.txt • Use your preferred editor to modify the file and save the changes • svn commit -m"Second commit comment" Test.txt <p>The commands should execute without any errors and without prompting the user for any extra input.</p>	

Notes:

- Your environment must be set up such that you can perform these operations without ever being prompted for user ID or password; for further information, please see the *Caching Client Credentials* topic in the official Subversion documentation

3.5.1.2.2.4 Subversion Under WINE-Crossover

The set up and use of Subversion with Enterprise Architect under Wine is almost identical to when running natively under Windows.

When running Enterprise Architect under Wine or CrossOver, you still use a Windows-based Subversion command line client.

There are some differences in the way you go about preparing your Subversion working environment, specifically in the way you install your Subversion client and how you check out a working copy folder from the Subversion repository.

Guide:

Topic	Detail	See also
System Requirements	<p>Sparx Systems has tested and succeeded in using Enterprise Architect with Subversion under Wine 1.2, on Mac OS 10.4 & 10.6.2, as well as Ubuntu 10.04.</p> <p>When using Wine 1.2 on the Ubuntu 9.10 platform, Sparx Systems was unable to successfully use the https: protocol to communicate with the SVN server, however we could use the svn: and file: protocols.</p>	
Installing a Subversion Client	<p>Wine is able to install applications from either a Windows . EXE file, or a .MSI installer file. You should place the installer for your Windows Subversion client in a convenient location on the native file system. Then, open a Wine console window from within Enterprise Architect and run the installer from within the Wine console. This ensures that your Subversion installation accesses the same C: drive and folders that Enterprise Architect is accessing.</p>	<p>Preparing a Subversion Environment Under Wine.^[259]</p>

Wine is able to install Subversion from either a Windows *.EXE* file, or a *.MSI* file. By performing your Subversion installation and initial check out from within a Wine console window that is opened from within Enterprise Architect, you can ensure that you have access to the same C: drive and folders that Enterprise Architect is accessing.

Use to:

- Set up Subversion for use with Enterprise Architect, running under Wine

How to:

Step	Action	See also
1	Start Enterprise Architect You do not have to load a project at this stage	
2	Select Tools Customize > Tools : New The Customize dialog displays, at the Tools page, and a new, blank entry is created	Custom Tools 
3	Define the new menu item entry, as follows: <ul style="list-style-type: none"> • In the newly-opened Menu contents field, type the name Wine Console • In the Command field, type wineconsole • In the Arguments field, type cmd • The Initial directory field should be left blank 	
4	Click on the Close button The Customize dialog closes	
5	Select Tools Wine Console A Wine console window opens	
6	Type C: and press (Enter) The Wine console switches to the C: drive	
7	Issue the command to install your Subversion client For example; <pre>C:\>/Installers/Subversion-client-1.6.12-1.win32.exe</pre> To install from a .msi file, use Wine's msiexec utility For example; <pre>C:\>msiexec "Slik-Subversion-1.6.9-win32.msi" /i</pre> Installation of the Subversion command line client begins	
8	Create a folder to serve as the working copy folder to be used by Enterprise Architect For example: <pre>C:\>mkdir C:\VC_workspaces\SVN_workcopy</pre>	
9	Issue the command to perform the initial check out from the Subversion repository, specifying the repository folder, the working copy folder, and the	

	<p>username and password</p> <p>(By specifying your Subversion username and password, you ensure that they are correctly cached by Subversion and available for use by Enterprise Architect)</p> <p>For example;</p> <pre>C:\>svn checkout https://myServer:8443/svn/repos_folder</pre> <pre>C:\VC_workspaces\SVN_workcopy --username myUserName --password myPassword</pre> <p>If the HTTPS protocol is specified when performing the initial Subversion check out, you are prompted to accept a security certificate; in this instance, press (P) to permanently accept the certificate</p> <p>The nominated local folder is configured as a Subversion working copy folder</p> <p>Any files already existing in the repository folder are downloaded to the working copy folder as working copy files</p>	
10	<p>Type Exit and press (Enter)</p> <p>The Wine console window closes</p> <p>You are now ready to load a project in Enterprise Architect and apply version control to it, following the normal Windows-based procedures</p>	<p>Version Control Setup [268]</p> <p>Use Version Control [275]</p>

Notes:

- You should copy the installer for your Windows Subversion client to a convenient location on the native file system, so that you can easily refer to it from within the Wine console window in step 7 above

3.5.1.2.2.5 TortoiseSVN

TortoiseSVN is a Windows shell extension for Subversion. TortoiseSVN provides icon overlays in Windows Explorer that are useful as a tool for observing the status of your Subversion controlled files.

It also enables you to create your repository sub-trees and check out local working copies from within Windows Explorer using simple menu commands.

TortoiseSVN can be downloaded from <http://tortoisesvn.net/downloads.html>.

Notes:

- Enterprise Architect must use the Subversion command line client. It **can not use** TortoiseSVN to communicate with the Subversion server

3.5.1.2.3 Create a CVS Environment

Before CVS can be used as a version control provider for Enterprise Architect, the appropriate software must be installed and configured by a CVS administrator.

The following topics describe some basic tasks that must be completed and some tools for performing these tasks, in order to create an operational CVS environment.

Topics:

Topic	Detail	See Also
Install server components	<p>Executable files for CVS can be obtained from the Free Software Foundation, or from March Hare Software.</p> <p>CVS server components are available to run on a wide range of different hardware and operating systems. Enterprise Architect is not affected by you choice of server components</p>	<p>FSF CVS</p> <p>March Hare CVS</p> <p>CVS Documentation</p>
Create a repository	Please consult the CVS documentation.	CVS Repository Administration
Create CVS users	Please consult the CVS documentation.	CVS Documentation
Create a new repository module	<p>It is good practice to create folders in your PC's file system for organizing your files and documents. Similarly, it is good practice to create a new repository module in CVS for each new Enterprise Architect model.</p> <p>A repository module represents a project, or set of related files in the repository.</p> <p>TortoiseCVS can greatly simplify the process of creating new repository modules.</p>	CVS Starting a New Project
Install client components	<p>Executable files for CVS can be obtained from the Free Software Foundation, or from March Hare Software.</p> <p>Enterprise Architect is a Windows based application - it requires a Windows based CVS command line client for integration.</p>	<p>FSF CVS</p> <p>March Hare CVS</p>
Create a working copy folder	<p>To create a local working copy, you must perform an initial check-out of a module from the CVS repository. This downloads a copy of the module, to create your local working copy.</p> <p>TortoiseCVS can greatly simplify the initial check out of a working copy folder.</p>	Create a CVS Local Workspace ^[262]
CVS under Wine/ CrossOver	The set up and use of CVS with Enterprise Architect under Wine is almost identical to when running natively under Windows, apart from some minor differences when installing the CVS client and performing the initial check out of the working copy folder.	
TortoiseCVS	TortoiseCVS is a Windows shell extension. Enterprise Architect can not use TortoiseCVS as its client, it must use the CVS command line client.	TortoiseCVS ^[264]

Notes:

- If you do not already use CVS for version control, you should consider using Subversion instead. Subversion's client-server protocols provide a broader range of possibilities for connecting to remote servers, with easier set up of secure connections
- Sparx Systems recommend that each new Enterprise Architect model being added to version control with CVS should have a separate repository module created for it, and users should create a new local working copy from the module to be used with that model

3.5.1.2.3.1 Prepare a CVS Local Workspace

In order to make use of CVS to provide version control of your model's packages, you must prepare a functional CVS working copy folder, that can be accessed through an Enterprise Architect version control configuration.

Use to:

- Prepare a CVS environment, suitable for use with Enterprise Architect

How to:

Step	Action	See also
1	Ask your system administrator to install CVS and create a remote repository with a module that you can use to control your Enterprise Architect package files. Your administrator must create a username and password for you before you can make a connection.	
2	Open a command prompt window and (if necessary create and then) navigate to a suitable directory to use as your CVS working copy directory. For example: <pre>C:\> mkdir myCVSWorkSpace C:\> cd myCVSWorkSpace</pre>	
3	Log in to the remote CVS repository. For example; <pre>C:\myCVSWorkSpace> cvs -d:pserver:myUserID@ServerName:/reposPath</pre> Replace <i>myUserID</i> with your CVS username, replace <i>ServerName</i> with the name of your CVS server and replace <i>reposPath</i> with the path to the repository on the server. A prompt for a password displays.	
4	Enter your password. You are logged in to the CVS server.	
5	Perform the initial check out of the CVS repository module, into the local working copy directory. For example; <pre>C:\myCVSWorkSpace> cvs -d:pserver:myUserID@ServerName:/reposPath</pre> (Replace <i>moduleName</i> with the name of the repository module that you want to check out.) A subdirectory is created in your current working directory, that has the same name as the module being checked out. Any files already existing in the repository module are downloaded to the working copy folder as working copy files.	
6	You should now verify that your CVS environment functions correctly.	Verify the CVS Workspace 263

Notes:

- Much of the process described above can also be performed (more simply) using the TortoiseCVS command "Make New Module"

3.5.1.2.3.2 Verify the CVS Workspace

After creating the CVS local working copy to hold the working copies of your Enterprise Architect package files, you should verify that it functions correctly.

You should be able to add files to CVS, and then commit changes to those files. You should also be able register as an editor of the file as well as retrieve the list of currently registered editors.

Use To:

- Verify correct operation of a newly created CVS working copy folder, before attempting to use it with Enterprise Architect

How To:

Step	Action	See also
1	Use Windows to open a command prompt window.	
2	Change directory to the one you specified as the working copy, in the cvs checkout command when preparing the CVS workspace. For example; <pre>C:\> cd myCVSWorkSpace</pre>	
3	Create a test file, such as Test.txt, containing the text "CVS Test". For example; <pre>C:\> echo CVS Test > Test.txt</pre>	
4	Execute the following CVS commands: <ul style="list-style-type: none"> • <code>cvs add Test.txt</code> • <code>cvs commit -m"Commit comment" Test.txt</code> • <code>cvs update Test.txt</code> • <code>cvs edit Test.txt</code> • <code>cvs editors Test.txt</code> <p>The commands should execute without any errors and without prompting the user for any extra input.</p> <p>The editors command should produce output that resembles the following;</p> <pre>Test1.txt myUserID Tue Aug 9 10:08:43 2009 GMT myComputer C</pre>	
5	Take note of the userID that follows the filename. Enterprise Architect must find and use this user ID when you create your version control configuration.	CVS Settings 1273

Notes:

- Your environment must be set up such that you can perform these operations without ever being prompted for input, such as user ID or password

3.5.1.2.3.3 TortoiseCVS

TortoiseCVS is a Windows shell extension for CVS. TortoiseCVS provides icon overlays in Windows Explorer that are useful as a tool for observing the status of your CVS controlled files.

It also enables you to create your repository modules and check out local working copies from within Windows Explorer using simple menu commands.

TortoiseCVS can be downloaded from: <http://www.tortoisecvs.org>

Notes:

- Enterprise Architect cannot use TortoiseCVS to communicate with the CVS server; it must use the CVS command line client

3.5.1.2.4 Create a TFS Environment

Before Enterprise Architect can be used with TFS, the appropriate software must be installed by a TFS administrator. Ask your system administrator to obtain and install the TFS server and client applications.

The following topics describe some basic tasks that must be completed in order to create a TFS environment that can be used by Enterprise Architect

Topics:

Topic	Detail	See Also
Obtain and install TFS	Enterprise Architect uses the TFS command line client to integrate TFS version control. It is assumed that the TFS command line client is installed as part of your Visual Studio installation.	
Choose a TFS project	It is good practice to create folders in your PC's file system for organizing your files and documents. Similarly, it is good practice to create a new TFS project, or least a new Source Control Folder within a project, for each new Enterprise Architect project. If you have a single Enterprise Architect project, that contains many different models, (for example, a DBMS hosted project with multiple model root nodes), you might choose to create a new TFS project for each separate model. Please consult your TFS documentation.	
Create a TFS workspace	The TFS workspace is used to map a local working folder on your PC, to a Source Control Folder within a TFS project. A single TFS workspace can map many different local folders - each one to a separate Source Control Folder. You can gain performance benefits for some of Enterprise Architect's version control operations if you keep your version controlled package files in a folder that is separate from other artefacts, such as source code files. This can be achieved either by creating a separate work space to use just for your package files, or by creating and mapping a separate folder for package files within an existing workspace.	TFS Workspaces [265]
Configure exclusive check-	The XML file format used for version control of Enterprise Architect's packages can not be merged like ordinary text files. Therefore,	TFS Exclusive Check Outs [266]

Topic	Detail	See Also
outs.	Enterprise Architect must enforce serialized editing of its version controlled packages. As a consequence, it is important that TFS is configured to use 'exclusive checkouts' for XML files.	

Notes:

- TFS can also be used with an SCC client. The MS TFS-SCC client is available for download from Microsoft's web site
- Visual Studio Integration (MDG Integration for Visual Studio 2005 or 2008) enhances TFS support by providing access to, for example, work items and bugs within both Enterprise Architect and the MDG Integration product

Learn More:

- [System Requirements](#)

3.5.1.2.4.1 TFS Workspaces

The TFS workspace is used to map a local working folder on your PC, to a Source Control Folder within a TFS project.

Use To:

- Associate a local folder on your PC, with a Source Control Folder in the TFS repository

How To:

Step	Action	See also
1	It is assumed a Team Project already exists, that you can use to control your Enterprise Architect package files. The following actions are carried out within MS Visual Studio.	
2	Connect to your TFS server. From the main menu, choose View Team Explorer .	
3	From the main menu, choose View Other Windows Source Control Explorer .	
4	From the main menu, choose Tools Connect to Team Foundation Server . A prompt displays for you to log in to the Team Foundation Server; another prompt then displays to choose an available Team Foundation Server.	
5	Choose an appropriate server and click OK . The selected Team Foundation Server is displayed in the Team Explorer pane. The server's project hierarchy is displayed in the Source Control Explorer's Folders pane.	
6	From the main menu, choose File New Team Project . The New Team Project wizard opens.	

7	Enter an appropriate name and description for the new Team Project. Choose the option to create an empty source control folder.	
8	Create a new Workspace.	
9	Create a new workspace. In the Source Control Explorer, click on the drop-down arrow in the Workspace field, then choose Workspaces The Manage Workspaces dialog is displayed.	
10	Click on the Add... button. The Add Workspace dialog is displayed.	
11	Enter an appropriate name for the new workspace and enter a comment if required.	
12	Click in the Source Control Folder column, then click the Browse button, to select a Source Control Folder. Select the new folder that was created in step 7.	
13	Click the Browse button in the Local Folder column and create a new local folder. This is the working copy folder into which Enterprise Architect exports the package files.	
14	Click OK . The new workspace is created and saved. The Add Workspace dialog closes.	
15	Click OK . The Manage Workspaces dialog closes.	

Notes:

- The local folder referenced in step 13, is the Working Copy Path that should be specified when defining an Enterprise Architect Version Control Configuration to use this TFS workspace

Learn More:

- [TFS Settings](#) ^[274]

3.5.1.2.4.2 TFS Exclusive Check Outs

The XML file format used for version control of Enterprise Architect's packages can not be merged like ordinary text files. Therefore, Enterprise Architect must enforce serialized editing of its version controlled packages. As a consequence, it is important that TFS is configured to use 'exclusive checkouts' for XML files. Otherwise, TFS can return file statuses that make it look like the package file is not checked-out by another user when indeed it is.

Use To:

- Configure TFS to enforce exclusive check outs for XML files

How To:

Step	Action	See also
1	Using Visual Studio, from the main menu select View Team Explorer .	
2	In the Team Explorer pane, right-click on the TFS Server name that is controlling the EA package files, then from the context menu select Team Foundation Server Settings Source Control File Types .	
3	Select the entry for XML files (or create an entry if necessary) then click on Edit .	
4	Clear the check-mark from the option Enable file merging and multiple check out .	

3.5.1.2.5 Create an SCC Environment

Before Enterprise Architect can be used with an SCC version control product, the appropriate software must be installed and configured by an administrator.

The following topics describe some basic tasks that must be completed in order to create an SCC based version control environment that can be used by Enterprise Architect

Topics:

Topic	Detail	See Also
Install and configure your chosen version control product.	<p>A version control server component is typically installed on a dedicated server machine. All Enterprise Architect users who require access to version control must be able to connect to the server machine.</p> <p>After installing the version control software, the administrator should also create version control user IDs for all users who require access to version control.</p> <p>Please consult the documentation for your particular version control product.</p>	<p>List of SCC compatible version control providers ^[253]</p>
Create a new SCC project	<p>It is good practice to create folders in your PC's file system for organizing your files and documents.</p> <p>Similarly, it is good practice to create a new version control project, or least a new folder within a project, for each new Enterprise Architect project.</p> <p>If you have a single Enterprise Architect project, that contains many different models, (for example, a DBMS hosted project with multiple model root nodes), you might choose to create a new version control project for each separate model.</p> <p>Please consult the documentation for your particular version control product.</p>	
Configure your SCC project to support exclusive check-outs for .XML files	<p>The XML file format used for version control of Enterprise Architect's packages can not be merged like ordinary text files. Therefore, Enterprise Architect must enforce serialized editing of its version controlled packages. As a consequence, it is important that your version control system is configured to use 'exclusive checkouts' for XML files.</p>	

Topic	Detail	See Also
Create a local working copy folder	<p>A working copy folder should be created on each users' machine, for Enterprise Architect to use when exporting and importing the version controlled package files.</p> <p>The working copy folder is the "sandbox" where you modify the controlled files. The working copy folder is usually associated with a folder that exists within the version control repository. Your version control product provides some means by which you associate a "working copy" folder with a repository folder.</p> <p>It is this folder that is specified as the Local Project Path, when defining your Version Control Configurations.</p> <p>Please consult the documentation for your particular version control product.</p>	Version Control Setup ^[268]

Notes:

- When installing the client component software on users' PCs, ensure that the SCC client is also installed, as it might not be a part of the default installation

3.5.1.3 Version Control Setup

Once the version control software has been installed and configured, to use your installed version control product you must first define a Version Control Configuration within Enterprise Architect.

Use to:

- Define a new Version Control Configuration, that can then be used to control the packages in your project

How to:

Step	Action	See also
1	Start Enterprise Architect and load the model for which you are defining the Version Control Configuration.	
2	Open the Version Control Settings dialog; Project Version Control Version Control Settings	Version Control Settings dialog ^[270]
3	Click on the New button.	
4	In the Unique ID field, type a suitable name.	
5	Select the Type of version control product you are connecting to, by clicking on the corresponding radio button.	
6	At this point, the middle section of the dialog changes to display a collection of fields specific to the type of Version Control Configuration you are defining. Enter details relating to the version control workspace that this configuration is to use.	
7	Click on the Save button. The new configuration is added to the Defined	

Step	Action	See also
	Configurations list.	
8	When you have finished defining your version control configurations, click on the Close button.	

Notes:

- Version Control Configuration details are stored in the user's Windows Registry settings, but each project stores a list of the configurations it uses, so that version control connections can be initialized as a project is being loaded
- You can define any number of Version Control Configurations for use in a single model, however any given package can be associated with only one configuration
- If you are using the Corporate or extended editions of Enterprise Architect with security enabled, you must also set up permissions to configure and use version control. See **List of Available Permissions** for more information

Learn More:

- [List of Available Permissions](#)^[205]

3.5.1.3.1 Re-use an Existing Configuration

Once a version control configuration has been defined for use in one project, it is possible to re-use that configuration in other projects.

Use To:

- Provide access to an already existing version control environment (a working copy directory and its associated repository that is already in use)
- Provide access to version controlled packages that were created (and version controlled) in another project

How To:

Step	Action	See also
1	Choose one of; <ul style="list-style-type: none"> • Project Version Control Version Control Settings • Right-click any package in the Project Browser Package Control Version Control Settings The Version Control Settings dialog displays.	Version Control Settings dialog ^[270]
2	Click on the New button. The various fields on the dialog are cleared, ready for data entry.	
3	In the Unique ID field, click on the drop-down arrow and select one of the previously-defined version control configurations. The details of the selected configuration are displayed in the dialog.	
4	Click on the Save button. The configuration details are saved and ready for use in the current project.	

3.5.1.3.2 Version Control Settings Dialog

The Version Control Settings dialog enables you to define Version Control Configurations, that are used by Enterprise Architect to communicate with your version control system.

Access: [Project | Version Control | Version Control Settings](#)

OR [Right-click on any package node | Package Control | Version Control Settings](#)

Use To:

- Define new Version Control Configurations
- Update existing Version Control Configurations

Reference:

Field	Usage	See Also
This mode is private	Specify whether this model database is to be accessed by just a single user (Private Model) or by multiple concurrent users (Shared Model). The default value is "unchecked" (Shared Model). If in doubt, use the default value.	
Save nested version controlled packages to stubs only	Specify whether the exported XML file for a version controlled package, shall contain package stubs (place holders) or full content for nested version controlled child packages. It is recommend to always export version controlled child packages as stubs only.	
Report cross package reference deletions before check-in	Choose whether or not Enterprise Architect should alert the user when cross package references are being removed from the exported XML package file, when compared to the previous revision of that file.	Report Deletion of Cross Package References ^[335]
Unique ID	A name that uniquely identifies the configuration. Enter a name to identify a new configuration, or use the drop-down list to retrieve details of a configuration previously defined in a different project.	
Type	Specify the type of version control system associated with this configuration. Once the type is selected, the middle section of the dialog changes to display a collection of fields relating to the type of Version Control Configuration you are defining. For a description of those fields, see the links at right. Set the type to SCC for; <ul style="list-style-type: none"> • MS Visual Source Safe • Rational Clear Case • Perforce • AccuRev • other SCC compatible clients For CVS, Subversion or TFS, choose the type that matches the product you are using.	SCC Settings ^[271] CVS Settings ^[272] SVN Settings ^[273] TFS Settings ^[274]

Field	Usage	See Also
New	Create a new Version Control Configuration.	
Save	Save the details of a new or updated configuration.	
Delete	Remove the definitions of the currently selected configuration from this model.	
Defined Configurations	List of configurations that are in use in the current model.	
In future, do not prompt for incomplete configurations	Choose whether or not the user is prompted to complete the definition of configurations that are not fully specified.	
Close	Close this dialog.	
Help	Display this Help topic.	

Notes:

- When you first open an Enterprise Architect project that was created by another user and that uses version control, the Version Control Configuration(s) used by that project do not yet exist in your Windows registry settings. You must complete the definitions of those configurations before you can use version control in that project
- It is important that for any given version controlled package file, all access of that file from any model and by any user, is done using Version Control Configurations having the same UniqueID
- If Project Security is enabled, you must have **Configure Version Control** permission to set up version control options for the current model
- It is possible to use multiple version control configurations in the same Enterprise Architect model

Learn More:

- [Applying version control to a package](#) ^[282]
- [Applying version control to a hierarchy of packages](#) ^[289]
- [Configure Version Control](#) ^[205]

3.5.1.3.2.1 SCC Settings

When you specify the type of your Version Control Configuration as SCC, the middle section of the Version Control Settings dialog changes to allow you to enter details that are specific to SCC based configurations.

Access: **Version Control Settings dialog > Choose type: SCC**

Use To:

- Define the working copy folder to be used with an SCC based version control configuration
- Specify the details necessary to connect to the SCC version control system

Reference:

Field	Usage	See Also
Local Project Path	Displays the full path of the folder that contains the local (working) copies of the XMI package files. This field is read-only, its value can only be set through	

Field	Usage	See Also
	use of the Select Path button.	
Select Path	<p>Choose the Local Project Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.</p> <ul style="list-style-type: none"> After choosing the appropriate folder path, the Select SCC Provider dialog opens. This dialog displays a list of all SCC providers that are installed on the current PC. Choose the SCC provider to use and click on the OK button At this point, the SCC client opens its own dialog to prompt you for information, such as the SCC Project to use You might also be asked to log in the the version control system At the conclusion of this process, all of the SCC details should be filled in. You can then save the definition by clicking Save on the Version Control Settings dialog 	System Requirements Version Control Settings dialog
Current User	<p>Displays the user name used to log on to the version control system that is accessed through this configuration.</p> <p>This field is read-only, the value it displays is retrieved from the SCC client.</p>	
SCC Provider	<p>Displays the name of the SCC provider.</p> <p>This field is read-only, the value it displays is retrieved from the SCC client.</p>	
SCC Project	<p>Displays the name of the SCC Project that this configuration attaches to.</p> <p>This field is read-only, the value it displays is retrieved from the SCC client.</p>	

Notes:

- After choosing an SCC Provider and clicking OK in the Select SCC Provider dialog, the SCC client takes control to prompt the user for information. SCC products implement this functionality in quite varied ways, but typically you are prompted to log in to the version control system, then prompted to choose the SCC Project to use and possibly a (server) folder contained within that project

3.5.1.3.2.2 CVS Settings

When you specify the type of your Version Control Configuration as CVS, the middle section of the Version Control Settings dialog changes to allow you to enter details that are specific to CVS based configurations.

Access: **Version Control Settings dialog > Choose type: CVS**

Use To:

- Define the working copy folder to be used with a CVS based version control configuration
- Specify the path to the CVS command line client

Reference:

Field	Usage	See Also
Working Copy Path	Displays the full path of the folder that contains the local (working) copies of the XML package files. This field is read-only, its value can only be set through use of the associated Select Path button.	
Select Path	Choose the Working Copy Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
Current User	This field is read-only. Its value is retrieved by Enterprise Architect, from a file named "CVS\Root", located in the folder specified by Working Copy Path above.	
CVS Exe Path	Displays the full path of the CVS command line client executable file. This field is read-only, its value can only be set through use of the associated Select Path button.	
Select Path	Specify the path to the CVS command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	

Notes:

- When connecting to a remote CVS repository, the **Current User** field should display the user name used to log into that repository. If this does not happen, it indicates that Enterprise Architect cannot extract the user name from the file "...WorkingCopyPath\CVS\Root" and the configuration does not work correctly

3.5.1.3.2.3 SVN Settings

When you specify the type of your Version Control Configuration as Subversion, the middle section of the Version Control Settings dialog changes to allow you to enter details that are specific to Subversion based configurations.

Access: **Version Control Settings dialog > Choose type: Subversion**

Use To:

- Define the working copy folder to be used with a Subversion based version control configuration
- Specify the path to the Subversion command line client

Reference:

Field	Usage	See Also
Working Copy Path	Displays the full path of the folder that contains the local (working) copies of the XML package files. This field is read-only, its value can only be set through use of the associated Select Path button.	

Field	Usage	See Also
Select Path	Choose the Working Copy Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
Subversion Exe Path	Displays the full path of the Subversion command line client executable file. This field is read-only, its value can only be set through use of the associated Select Path button.	
Select Path	Specify the path to the Subversion command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	

3.5.1.3.2.4 TFS Settings

When you specify the type of your Version Control Configuration as TFS, the middle section of the Version Control Settings dialog changes to allow you to enter details that are specific to TFS based configurations.

Access: **Version Control Settings dialog > Choose type: TFS**

Use To:

- Define the working copy folder to be used with a TFS based version control configuration
- Specify the user name and password to log in to the TFS server
- Specify the path to the TFS command line client

Reference:

Field	Usage	See Also
Working Copy Path	Displays the full path of the folder that contains the local (working) copies of the XML package files This field is read-only, its value can only be set through use of the associated Select Path button	
Select Path	Choose the Working Copy Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
Server Name	Displays the name of the TFS Server that is associated with the working copy folder specified in Working Copy Path above. This field is read-only. Enterprise Architect retrieves the value it displays by querying the TFS client.	
Workspace Name	Displays the name of the TFS Workspace that is associated with the working copy folder specified in Working Copy Path above. This field is read-only. Enterprise Architect retrieves the value it displays by querying the TFS client.	
User Name	Specify the user name with which to log into the TFS Server.	

Field	Usage	See Also
Password	Specify the password with which to log into the TFS Server.	
TFS Exe Path	Displays the full path of the TFS command line client executable file. This field is read-only, its value can only be set through use of the associated Select Path button.	
Select Path	Specify the path to the TFS command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	

Notes:

- Users who automatically log in to TFS through means external to Enterprise Architect (for example, through MS Integrated Security) can leave the User Name and Password fields blank
- If the Password field is blank, Enterprise Architect retrieves the current user's Windows username and uses that value when determining whether a package file is checked out to them or to some other user
- TFS version control can also be accessed using the TFS MSSCCI client. To make use of the TFS MSSCCI client, please define an SCC based Version Control Configuration

3.5.1.4 Use Version Control

Once your version control product is installed and a suitable environment has been created, Enterprise Architect can then utilize that environment to control your project's packages.

The following topics describe use of the version control features of Enterprise Architect, accessed through the Package Version Control Menu.

Guide:

Topic	Detail	See also
Version Control Settings	Version Control Configurations are used by Enterprise Architect to communicate with your version control system. You must define a Version Control Configuration in your project and then use that configuration to control the packages in your project.	Version Control Setup [268]
Configure a Package	To put a package under version control, you must mark the package as a controlled package, specify the version control configuration to control it and associate an XML file with the package.	Configure Controlled Package [282]
Check In a Model Branch	The Check In a Model Branch command allows you to check-in all packages involved in a particular unit of work, as a single operation.	Check In a Model Branch [283]
Check Out a Model Branch	The Check Out a Model Branch command allows you to check-out all packages within a selected model branch, as a single operation.	Check Out a Model Branch [283]
Check In a Package	The Check In command allows you to check-in the package currently selected in the Project Browser.	Check In a Package [283]

Topic	Detail	See also
Check Out a Package	To work on a version controlled package you must have the package checked out. The Check Out command allows you to check-out the package currently selected in the Project Browser.	Check Out a Package ^[285]
Undo Check Out of a Package	The Undo Check Out command allows you to undo the check-out of a package, discarding any modifications that have been made by restoring the package content to the latest revision held in version control.	Undo Check Out of a Package ^[286]
Import a Package From Version Control	You can retrieve packages from version control that have been created by other users, or by you in another model, and import them into your current model.	Include Other Users Packages ^[289]
Apply Version Control to a Model Branch	Enterprise Architect enables you to apply version control to all packages within a selected model branch, in a single operation. In this context, a model branch is a package that is currently selected in the Project Browser, and all of the packages contained within it.	Apply Version Control to Branches ^[289]
Export a Version Controlled Model Branch	The Export a Model Branch command exports version control information about the root package of a model branch, that is used to simplify the process of exporting and importing a hierarchy of packages from one model to another.	Export Controlled Model Branch ^[290]
Import a Model Branch From Version Control	Model branch files can simplify the process of exporting and importing a hierarchy of packages from one model to another. The Import a Model Branch command uses Enterprise Architect's Model Branch Files, of which there are few, to retrieve information about the root package file and import the model branch.	Import Controlled Model Branch ^[291]
View Package Revision History	The File History command allows you to review the change history of version controlled packages. It also enables you to check out a prior revision of the package for editing, effectively allowing you to roll-back to a prior revision of the package.	Review Package History ^[293]
Validate Package Configurations	You can test the validity of the version control settings associated with each version controlled package within your current model.	Validate Package Configurations ^[296]
Resynchronize the Status of Version Controlled Packages	Enterprise Architect enables you to re-synchronize the version control status of version controlled packages within your project with the status reported by your version control provider.	Resynchronize Package Status ^[297]

Notes:

- Database replication should not be combined with version controlled packages
- If the packages under version control contain any alternative images and those images are subject to frequent change, you can set the **Export alternate images** option on the **Options** dialog to export the images to the version control repository when you check in the packages. If the images are not subject to frequent change, do not select this option and instead use **Export / Import Reference Data** to manage alternative images

3.5.1.4.1 Project Version Control Menu

Provides access to various project related version control functions

Access: **Project | Version Control**

Use to:

- Apply version control to a package
- Modify or check version control settings
- Validate settings for all version controlled packages
- Resynchronize the status of all version controlled packages, with their associated version control system.
- Allow working when not connected to version control

Reference:

Action	Usage	Shortcut	See also
Configure Current Package	Display the Package Control Options dialog, which enables you to specify whether this package (and its children) is version controlled, and which version control configuration applies.	Ctrl + Alt + P	Package Control Options ^[287] dialog
Version Control Settings	Display the Version Control Settings dialog ^[270] .		Version Control Settings dialog ^[270]
Validate Package Configurations	Test the validity ^[296] of the version control settings associated with each version controlled package within your current model.		Validate package configuration ^[296]
Re-Synch Statuses of All Packages	Resynchronize the version control status of packages ^[297] as recorded in your Enterprise Architect project when they are out of synchronization with the version control status reported by your version control provider. The function acts on all version controlled packages within the Enterprise Architect project, updating the values recorded in the project to match the values reported by the version control provider, without performing any XML import or export.		Resynchronize the version control status of packages ^[297]
Work Offline	Work independently of the version control server. By selecting Work Offline before loading your project, you can prevent Enterprise Architect from attempting to connect to a version control server that is unavailable, thereby avoiding the delays of waiting for the server to respond and any associated error messages.		Offline version control ^[250]

3.5.1.4.2 Package Version Control Menu

The Package Control menu, when invoked from a version controlled package, provides access to all of the version control operations that are performed on packages.

Access: **Right-click on a version controlled package node | Package Control**

Use to:

- Open the dialog for working with baselines
- Apply version control to a single package or a selected hierarchy of packages
- Check-in and check-out single packages or a selected hierarchy of packages
- Update packages to the latest revision from the version control repository
- Inspect the revision history or properties of the XML file associated with a package
- Revert a package to a previous revision
- Compare the current model content of a package, against the latest revision of the package in version control
- Import and export hierarchies of packages (model branches) to and from the model, through the version control system
- Synchronize the status of a package, with the version control system

Reference:

Action	Usage	Shortcut	See also
Configure	Display the Package Control Options dialog for the currently selected package Use to apply or remove version control for a package (specify version control settings) or to specify a file for use in XML package control	Ctrl + Alt + P	Package Control Options Dialog ^[282]
Manage Baselines	Create a Baseline of the current package, or compare the current package with a previous Baseline	Ctrl + Alt + B	Create a Baseline ^[31]
Check In Branch	Check-in packages contained in the currently selected model branch For the selected branch of the model, (that is, the selected package and all of its child packages) display the Select Packages to Check In dialog, listing all version controlled packages within that branch that are checked out to you; you can then select packages in the displayed list, to be submitted for check-in You can also choose to keep the packages checked-out after committing a new revision to version control		Check In a Model Branch ^[283]
Check Out Branch	Recursively check out all packages contained within the currently selected model branch, that are version controlled and checked-in (that is, the selected package and all of its child packages)		Check Out a Model Branch ^[284]

Action	Usage	Shortcut	See also
Check In	<p>Commit a new revision of the currently selected package to the version control repository and lock the package against further editing</p> <p>Only available for packages that are checked-out by the current user</p>		Check In a Package ^[285]
Check Out	<p>Synchronize the currently selected package with the latest revision from the version control repository and unlock the package to allow editing</p> <p>Only available for packages that are not already checked-out (and whose associated package file is not checked-out)</p>		Check Out a Package ^[285]
Undo Check Out	<p>Restore the selected package to the latest revision in the version control repository and lock the package against further editing</p>		Undo Check Out of a Package ^[286]
Put Latest	<p>Commit a new revision of the currently selected package to the version control system, while keeping the package checked-out</p> <p>This is equivalent to checking a package in and immediately checking it back out again</p> <p>Only available for packages that are checked-out by the current user</p>		
Get Latest	<p>Synchronize the currently selected package with the latest revision from the version control repository</p> <p>Available only for packages that are checked in</p>		Update Selected Package ^[287]
Get All Latest	<p>Update all of the version controlled packages in the project, to the latest revision retrieved from version control</p> <p>Only updates packages that are currently checked in</p> <p>Once the latest revisions are retrieved, Enterprise Architect scans all the controlled packages and fixes any missing cross-references by comparing the package with its XML 1.1 file</p> <p>If the cross-reference information in the XML does not match the model, Enterprise Architect updates the model with the information from the XML and records this update in the Output window</p> <p>You can roll back such updates by</p>		Update All Packages ^[288] Output Windows ^[127]

Action	Usage	Shortcut	See also
	<p>selecting the entry in the Output Window and using the context menu option Rollback Update (or Rollback Selected Updates if multiple entries are selected)</p> <ul style="list-style-type: none"> • Closing the model clears the entries in the Output window • An entry in the Output window is also cleared as and when you roll-back the update for it 		
<p>Scan XMI and Reconcile Model</p>	<p>Scan the package XMI files associated with each of the project's controlled packages and restore any diagram objects or cross-references that are detected as missing from the project</p> <p>This function is useful in team environments where each user maintains their own private copy of the model database (i.e. multiple private EAP files) and model updates are propagated through the use of controlled packages; it provides no benefit when the model is hosted in a single shared database that is accessed by all team members</p> <p>Each controlled package is compared with its associated XML file and, if the cross-reference information in the model does not match the XML, Enterprise Architect updates the model with the information from the XML and records the update in the Output window</p> <p>You can roll back such updates by right-clicking on the entry in the Output window and selecting the context menu option Rollback Update (or Rollback Selected Updates if multiple entries are selected)</p> <p>Closing the model clears the entries in the Output window; an entry in the Output window is also cleared as and when you roll-back the update for it</p> <p>This functionality is invoked automatically as part of the Get All Latest operation</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke Scan XMI and Reconcile Model from time to time, following the re-importation of controlled packages - for example, after using Get Latest to update a number of packages - or after performing a number of package check-outs</p> <ul style="list-style-type: none"> • As a general rule, avoid running 		<p><i>Version Control Best Practices</i> white paper http://sparxsystems.com/WhitePapers/Version_Control.pdf</p>

Action	Usage	Shortcut	See also
	<p>Scan XMI and Reconcile Model while you have uncommitted changes in your model</p> <ul style="list-style-type: none"> • Generally, you should: <ul style="list-style-type: none"> • Check-out a number of packages • Invoke Scan XMI and Reconcile Model • Make your modifications • Commit any outstanding changes before you check-out more packages and run Scan XMI and Reconcile Model again 		
File Properties	Display version control properties pertaining to the XMI export file associated with the currently selected package; this also identifies who has checked out the package		
File History	<p>Display change history information for the currently selected package</p> <p>Revert to or check-out a prior revision of the package</p>		Review Package History ^[293]
Compare with controlled version	Compare the currently selected package with the latest revision of its associated XMI file retrieved from version control		
Add Branch to Version Control	<p>Apply version control to all packages within a selected <i>model branch</i>, in a single operation</p> <p>In this context, a model branch is a package that is currently selected in the Project Browser, and all of the packages contained within it</p>		Apply Version Control to Model Branches ^[289]
Export as Model Branch	Export a newly created model branch from your own private copy of a model		Exporting VC branches ^[290]
Import a Model Branch	Retrieve a model branch and import it into either the source model or another model		Importing VC branches ^[291]
Get Package	Access packages in the version control repository that are not currently available in your model		Including Other User's Packages ^[289]
Re-synch Status With VC Provider	<p>Update the version control status value recorded for the selected package in the Enterprise Architect project to match the value reported by the version control provider, without performing an XMI import or export</p> <p>Use this function when the package's version control status recorded in your Enterprise Architect project is out of</p>		Match the value reported by the version control provider ^[297]

Action	Usage	Shortcut	See also
	synchrony with the version control status reported by your version control provider		
Version Control Settings	Display the Version Control Settings dialog		Version Control Settings dialog ^[270]
Update Package Status	Provide a bulk update on the status of a package, including status options such as Proposed , Validate and Mandatory This command is not specific to version controlled packages		

Notes:

- If the selected package is not under version control, this menu displays a number of different options

Learn More:

- [Controlled Package Menu](#)^[328]

3.5.1.4.3 Configure Controlled Package

To put a package under version control, you must mark the package as a controlled package, specify the version control configuration to control it and associate an XML file with the package. Enterprise Architect can then export and import the package data to and from the file as required and issue commands to the version control system as appropriate.

Use to:

- Apply version control to a single package

How to:

Step	Action	See also
1	In the Project Browser, right-click on the package to be placed under version control and choose Package Control Configure... The Package Control Options dialog displays.	
2	Click on the Control Package checkbox to insert a check mark, indicating that this package is to be controlled.	
3	Click on the Version Control drop-down arrow and select the Version Control Configuration to be used to control this packages. The XMI Filename field then displays a default filename for the package export file, based on the package name.	
4	Optionally, modify the default XMI Filename . Choose either; <ul style="list-style-type: none"> • type a new name into the filename field • click on the browse button (...) to open a file chooser dialog. The target file must be located within the working copy folder of the selected	

Step	Action	See also
	Version Control Configuration, or one of its sub-folders.	
5	Optionally, update the Version ID for this package. The Version ID field defaults to 1.0; if necessary, change this to the appropriate reference.	
6	Optionally, update the Owner field. The Owner field defaults to your user name; if necessary, type or select the name of the user who owns the package.	
7	Click on the OK button. The Add Package to Version Control dialog displays.	
8	Optionally, clear the Keep checked out checkbox. After applying version control, the package is either checked-out for editing, or checked-in and locked against editing, in accordance with this setting.	
9	Click on the OK button. The Add Comment dialog displays.	
10	Optionally, add any further comments to the default comment. Enterprise Architect provides a default comment that includes the current date & time.	
11	Click on the OK button. The current package is exported to the nominated XML file, which is then committed to version control. The package icon in the Project Browser is updated to reflect the package's version control status.	Project Browser Indicators [249]

Notes:

- If you are using the Corporate or extended editions of Enterprise Architect with security enabled, then this feature is only available to users who have been granted permission to configure and use version control

Learn More:

- [List of Available Permissions](#) [205]

3.5.1.4.4 Check In a Model Branch

The Check In a Model Branch command enables you to check-in all packages involved in a particular unit of work, as a single operation. You can even commit new revisions of the affected packages as you complete milestones, while still keeping the packages checked-out for further editing.

Use to:

- Check in a selection of packages from within the currently selected model branch

How to:

Step	Action	See also
1	<p>Right-click on the package icon at the root of the model branch that is to be checked in and select Package Control Check In Branch.</p> <p>The Select Packages to Check-in dialog displays, listing all version controlled and checked-out packages within that model branch.</p>	Check In Package ^[285] Check Out Model Branch ^[284]
2	<p>Select the packages to be checked-in. (By default, the entire list is selected.)</p> <p>Optionally;</p> <ul style="list-style-type: none"> • Click an individual package to select just that package • Ctrl + click to add or remove an individual package from the selection • Shift + click to select a range of packages • Click on All to select all listed packages • Click on None to clear the selection 	
3	<p>Optionally, tick the checkbox Keep packages checked-out after committing new revision.</p> <p>Use this option to commit into version control, a new revision of all selected packages, while keeping all packages checked out for further editing.</p>	
4	<p>Click on the OK button.</p> <p>The Add Comment dialog displays. A default comment is provided that contains the current date and time.</p>	
5	<p>Optionally, modify the default check-in comment.</p> <p>This comment is applied to all packages that are checked in.</p>	
6	<p>Click on the OK button.</p> <p>The selected packages are exported and checked-in. The package icons are updated to reflect any change in version control status. If you opted to keep packages checked out, there is no change in status.</p>	

3.5.1.4.5 Check Out a Model Branch

The Check Out a Model Branch command allows you to check-out all packages within a selected model branch, as a single operation.

Use to:

- Check out an entire sub-tree of model packages

How to:

Step	Action	See also
1	<p>Right-click on the package icon at the root of the model branch that is to be checked out and select Package Control Check Out Branch.</p> <p>The selected package and all of its contained sub-packages are recursively checked out.</p> <p>Any packages that cannot be checked-out are listed in a message box, with</p>	Check Out Package ^[285] Check in Model Branch ^[283]

Step	Action	See also
	a brief description of the problem. For example: <i>The package is already checked out by user 'Fred'.</i>	
2	When Project Security is enabled in <i>Lock to Edit</i> mode, Enterprise Architect prompts you to apply a User Lock throughout the selected model branch before proceeding.	

Notes:

- Any packages that cannot be checked-out are listed in a message box, with a brief description of the problem. For example: The package is already checked out by user 'Fred'

3.5.1.4.6 Check In a Package

The Check In command allows you to check-in the package currently selected in the Project Browser.

Use to:

- Check in a single package

How to:

Step	Action	See also
1	In the Project Browser, right-click on the package that is to be checked in and select Package Control Check In . The package is exported and the Add Comment dialog displays. A default comment is provided that contains the current date and time.	Check In Model Branch ^[283] Check Out Package ^[285]
2	Optionally, modify the default check-in comment.	
3	Click OK . The package file is checked-in to version control and the package icon is updated to reflect the change in version control status.	

3.5.1.4.7 Check Out a Package

To work on a version controlled package you must have the package checked out. When a package is checked out to a specific user, no other user can check out the package to make changes to it until it has been checked in again.

The Check Out command allows you to check-out the package currently selected in the Project Browser.

Use to:

- Check out a single package

How to:

Step	Action	See also
------	--------	----------

1	<p>In the Project Browser, right-click on the package that is to be checked out and select Package Control Check Out.</p> <p>The local XML file associated with the package is checked-out from version control.</p>	<p>Check Out Model Branch^[284]</p> <p>Check In Package^[285]</p>
2	<p>When working in a Private Model, if Enterprise Architect determines that the package content in the model is already up to date with the latest revision of the package file retrieved from version control, then the Import Package dialog displays. This dialog is not displayed for Shared Models.</p> <p>The following options are offered;</p> <ul style="list-style-type: none"> • Force reload From XML • Accept current package <p>Choose Accept current package to skip the process of re-importing the package from XML.</p> <p>The option Refresh model view allows you to specify whether or not to refresh the Project Browser and diagrams, by reloading the package content from the project database.</p> <p>The Import Package dialog also has an option Always use these settings. When selected, if you subsequently check out a package that is found to be up to date, the same settings are applied again without displaying the dialog.</p> <ul style="list-style-type: none"> • To display the dialog if, for example, you want to change the settings, press a Ctrl key while you select the Package Control Check Out menu option. 	
3	<p>The package file is imported into your model.</p> <p>The package icon is updated to reflect the change in the package's version control status.</p>	

Notes:

- If you check out a version controlled package whilst offline, the package icon has a red figure 8 in front of it

Learn More:

- [Offline Version Control](#)^[250]

3.5.1.4.8 Undo Check Out of a Package

The Undo Check Out command allows you to undo the check-out of a package, discarding any modifications that have been made by restoring the package content to the latest revision held in version control. The package returns to a Checked-in state and subsequently can be checked out by any user.

Use to:

- Undo the Check out of the selected package, restoring the package content to the latest revision held in version control

How to:

Step	Action	See also
------	--------	----------

1	<p>In the Project Browser, right-click on the package and select Package Control Undo Check Out.</p> <p>A confirmation dialog is displayed.</p>	
2	<p>Click OK.</p> <p>The latest revision of the package is retrieved from version control and re-imported into your model.</p> <p>The package icon is updated to reflect the change in the package's version control status.</p>	

3.5.1.4.9 Update to the Latest Revision of Selected Package

When you are part of a team working in a Distributed Model environment, from time to time, you will want to update your model with the changes that other team members have committed into version control. The Get Latest command allows you to bring in updates made by other users, for the package currently selected in the Project Browser.

Use to:

- Update a single package to the latest revision retrieved from version control

How to:

Step	Action	See also
1	<p>In the Project Browser, right-click on the package that is to be updated and select Package Control Get Latest.</p> <p>The local XML file associated with the package is updated to the latest revision from version control. The XML file is imported into your model database, updating the package in your model.</p>	<p>Update All Packages [288]</p>
2	<p>When working in a Private Model, if Enterprise Architect determines that the package content in the model is already up to date with the latest revision of the package file retrieved from version control, then the Import Package dialog displays. This dialog is not displayed for Shared Models.</p> <p>The following options are offered;</p> <ul style="list-style-type: none"> • Force reload From XML • Accept current package <p>Choose Accept current package to skip the process of re-importing the package from XML.</p> <p>The option Refresh model view allows you to specify whether or not to refresh the Project Browser and diagrams, by reloading the package content from the project database.</p> <p>The Import Package dialog also has an option Always use these settings. When selected, if you subsequently check out a package that is found to be up to date, the same settings are applied again without displaying the dialog.</p> <ul style="list-style-type: none"> • To display the dialog if, for example, you want to change the settings, press a Ctrl key while you select the Package Control Get Package menu option. <p>The package file is imported into your model, in accordance with your</p>	

	selection.	
--	------------	--

Notes:

- The Get Latest command is disabled for any package that is checked-out (to anybody) in the currently loaded project.
- When using a Shared Model environment, where all users are connected to a single model database, you should reload the package from the database, rather than using the Get Latest command.

3.5.1.4.10 Update to the Latest Revision of All Packages

When you are part of a team working in a Distributed Model environment, from time to time, you will want to update your model with the changes that other team members have committed into version control. The command Get All Latest, allows you to bring in updates made by other users, for all of the version controlled packages in the currently loaded project.

Use to:

- Update all packages in your project, to the latest revision retrieved from version control

How to:

Step	Action	See also
1	<p>In the Project Browser, right-click on any package and select Package Control Get All Latest.</p> <p>All of the local XML files for all of the version control configurations used in the project, are updated to the latest revision from version control.</p> <p>Enterprise Architect scans the packages in the model, to determine which ones are up to date and which are not, compared to the latest revisions of the associated package files.</p> <p>You are then prompted to choose one of the following import options for packages that are up to date;</p> <ul style="list-style-type: none"> • Import changed files only, • Always import • Prompt for each file. 	<p>Update Selected Package^[287]</p> <p>See the note below.</p>
2	<p>Click on the OK button.</p> <p>The version controlled packages in your project are updated as appropriate.</p>	

Notes:

- There is no need to re-import packages that are already up to date. Re-importing packages involves first deleting them from the project, before re-importing them from the XML file. This is quite time consuming as well as quite unnecessary. We strongly recommend using the default option "Import changed files only".
- The Get All Latest command will not update any package that is checked-out (to anybody) in the currently loaded project. Otherwise, any changes not yet committed to version control would be discarded.
- When using a Shared Model environment, where all users are connected to a single model database, there is no need to use the Get All Latest command. The information in the model database is always ahead of what is committed into version control. You should from time to time refresh your view of the model database, by reloading diagrams or reloading package content in the Project Browser.

3.5.1.4.11 Include Other Users' Packages

You can retrieve packages from version control that have been created by other users, or by you in another model, and import them into your current model.

Other users might be creating packages to use in your model. If you are not sharing a SQL database or .eap file, those packages do not automatically become part of your model. If the packages have been placed into version control, you can retrieve them and import them into your model as children of an existing package, using the **Get Package** command.

Use to:

- Retrieve and import packages from version control, into the current project

How to:

Step	Action	See also
1	In the Project Browser, right-click on the package to use as the parent of the incoming package	
2	Select the Package Control Get Package context menu option. The Get Shared File dialog displays.	
3	Click on the drop-down arrow of the Version Control Configuration field, and select the version control configuration associated with the package to retrieve. The file list is populated with the names of files available through that configuration, for retrieval and import into your model.	
4	Select the package file to import into your model and click on the OK button. The selected package file is imported as a new child package, under the parent package selected in step 1.	

Notes:

- You must have access to the package files through the version control system and you must define a Version Control Configuration through which to access those files
- The version control configuration must use the same unique ID that was originally used to add the package to version control
- XML package files associated with packages that are already part of your project, are NOT included in the list of files available for import

3.5.1.4.12 Apply Version Control To Branches

Enterprise Architect enables you to apply version control to all packages within a selected *model branch*, in a single operation. In this context, a model branch is a package that is currently selected in the **Project Browser**, and all of the packages contained within it.

Use to:

- Apply version control to all packages within a selected model branch

How to:

Step	Action	See also
1	Right-click on the required package and select the Add Branch to Version Control context menu option. The Apply VC to Branch dialog displays.	
2	Click on the drop-down arrow in the Version Control Configuration field and select the configuration to use.	
3	Optionally, tick the checkbox Export as Model Branch . Once the version control operation is complete Enterprise Architect creates a Model Branch file (.EAB file) for this branch.	Export Model Branch 290 Import Model Branch 291
4	Click on the OK button. Enterprise Architect creates a number of sub-folders within the version control working copy folder, then exports all of the packages within the selected model branch. Enterprise Architect generates filenames for the XML files, based on the package GUIDs.	

Notes:

- The Version Control Configuration to be used in this operation must be defined within the model before selecting this command
- When invoked on the model root node, this command applies version control to every package within the model

3.5.1.4.13 Export Controlled Model Branch

Model branch files can simplify the process of exporting and importing a package hierarchies from one model to another.

You might want to export a newly created model branch from your own private copy of a model so that, for example:

- Another user can import that branch into their own private copy of the same model
- It can be imported for inclusion as a common branch in a number of different models

Applying version control to an Enterprise Architect model can result in many XML files placed under version control. It could then be hard to locate and import the file corresponding to the root of a particular model branch. Enterprise Architect's Model Branch Files (.EAB files) overcome this problem by simplifying the retrieval of model hierarchies for use in other models.

Use to:

- Create a Model Branch File to represent a package hierarchy that is stored in version control

How to:

Step	Action	See also
1	Select the version controlled package to export as a model branch.	
2	Right-click on the package and choose Package Control Export as Model Branch . The Export as Model Branch dialog displays.	Import Model Branch 291

3	<p>In the EAB Filename field, type a name for your Model Branch File.</p> <p>Alternatively, click on (...) and browse for the file location.</p> <p>Note that the package name is supplied as a default.</p>	
4	<p>Click OK.</p> <p>A branch file is created to represent the selected package. The branch file is committed to version control using the same Version Control Configuration that controls the package selected in step 1.</p>	

Notes:

- You can specify any file name, including sub-folder names, as long as the file is contained in or below the working folder of your version control configuration
- The facility is only enabled for packages that are already under version control

3.5.1.4.14 Import Controlled Model Branch

Applying version control to an Enterprise Architect model can result in many XML files placed under version control. It could then be hard to locate and import the file corresponding to the root of a particular model branch. Enterprise Architect's Model Branch files overcome this problem by simplifying the retrieval of package hierarchies for use in other models.

The Import a Model Branch command uses Enterprise Architect's Model Branch Files, of which there are few, to retrieve information about the root package file and import the model branch. The Model Branch File records information such as the name and type of the version control configuration for the selected package, and the relative filename of the version controlled XML file associated with the package.

Use to:

- Retrieve a model branch created by another user in a private copy of a model, to import it into your own private copy of the same model
- Retrieve a model branch that is common in many models, for inclusion in a new model

How to:

Step	Action	See also
1	<p>Ensure that you have an operational version control environment that can be accessed by Enterprise Architect and that you have downloaded all of the version controlled package files and the model branch file associated with the model branch to import, into a valid and accessible working copy folder.</p>	Version Control Product Setup ^[252]
2	<p>Select the package into which the model branch is to be imported.</p>	
3	<p>Right-click on the package and choose Package Control Import a Model Branch.</p> <p>The Import VC Model Branch dialog displays.</p>	
4	<p>Choose one of either;</p> <ul style="list-style-type: none"> • Use the lower portion of the Import VC Model Branch dialog to select a model branch file. <p>(This is the simpler option if the associated version control configuration has already been saved in the current model. Continue with step 5.)</p>	Manually Locating Model Branch Files ^[292]

	<ul style="list-style-type: none"> Click on the Find a Model Branch (.EAB) file button. <p>(This option is useful when you have not yet defined the version control configuration that is associated with the model branch to be imported.)</p> <p>If choosing this option, please refer to "Manually Locating Model Branch Files".</p>	
5	<p>Click the drop-down arrow in the Select a Version Control Configuration field and select the configuration</p> <p>A list of EAB files controlled by that configuration is displayed.</p>	
6	<p>Select the Model Branch File from the Select a Model Branch (.EAB) file list.</p>	
7	<p>Click OK.</p> <p>Enterprise Architect imports the root package specified in the Model Branch File and recursively imports and populates all the sub-packages contained in the root package.</p>	

Notes:

- The **Import a Model Branch** command is only enabled for packages that you (the current user) are able to edit, as the imported model branch is inserted into the model under the selected package

Learn More:

- [Export a Model Branch](#)^[290]

3.5.1.4.14.1 Manually Locating Model Branch Files

When importing a Model Branch File from version control, sometimes you do not yet have the associated version control configuration saved in the model that is receiving the import. In this situation, it is simpler to manually locate the Model Branch File and let Enterprise Architect derive the details of the configuration from the branch file you select.

Use To:

- Browse the file system to locate the Model Branch File (.EAB) to import.
- Let Enterprise Architect determine the details of the required version control configuration, based on information contained in the Model Branch File

Access: **Right Click on a package | Package Control | Import a Model Branch > Find a Model Branch (.EAB) file**

How to:

Step	Action	See also
1	Ensure that you have an operational version control environment that can be accessed by Enterprise Architect and that you have downloaded all of the version controlled package files and the model branch file associated with the model branch to import, into a valid and accessible working copy folder.	Version Control Product Setup ^[252]

2	Choose the package in the Project Browser, into which the model branch is to be imported.	
3	Right-click on the package and choose Package Control Import a Model Branch . The Import VC Model Branch dialog displays.	
4	Click on the button Find a Model Branch (.EAB) file . A file selection dialog is displayed.	Import Controlled Model Branch ^[291]
5	Browse for and select the Model Branch File that represents the model branch to import.	
6	Click Open . If the version control configuration referenced by the file is fully defined within the current model, the import commences at this point. Otherwise, Enterprise Architect displays a dialog prompting you to complete the required configuration.	
7	Click Yes to proceed with completing the definition of the version control configuration. The Version Control Settings dialog is displayed.	Version Control Settings dialog ^[270]
8	Complete the definition of the configuration. (Typically this involves simply specifying the working copy folder.)	
9	Click Save . The configuration details are saved. The import of the model branch proceeds.	

Notes:

- The Import a Model Branch command is only enabled for packages that you (the current user) are able to edit, as the imported model branch is inserted into the model under the selected package
- It is not necessary to have the relevant Version Control Configuration set up within Enterprise Architect before issuing this command - Enterprise Architect prompts you to complete specification of the configuration if necessary

Learn More:

- [Export a Model Branch](#)^[290]

3.5.1.4.15 Review Package History

The File History command allows you to review the change history of version controlled packages.

It also enables you to check out a prior revision of the package for editing, effectively allowing you to roll-back to a prior revision of the package.

Use to:

- Review the change history of version controlled packages
- Revert to a previous revision of a version controlled package

How to:

Step	Action	See also
1	In the Project Browser, select the package for which to view change history.	
2	Right-click on the package and choose Package Control File History . For version control environments using Subversion, CVS or TFS command line clients, Enterprise Architect's File Version History dialog displays. For version control environments using SCC based clients, your particular product opens its own File Version History dialog	Review Package History - SCC Client [294]
3	Click on a revision number in the Revisions field, to select that revision and view its log entry.	
4	Optionally, click on either; <ul style="list-style-type: none"> • Check Out • Retrieve <p>If you click on Check Out, the selected revision of the package file is retrieved from version control and imported into your model as a package that is checked out for editing. You can subsequently check in this revision as a new HEAD revision - effectively allowing you to revert a package to a prior revision.</p> <p>If you click on Retrieve, the selected revision of the package file is retrieved from version control and imported into your model, but the package remains flagged as checked-in, so it cannot be modified. Subsequently checking out the package updates the package to the latest revision before it is unlocked for editing.</p>	

Notes:

- If the selected package was already checked out in the current model, the **Retrieve** and **Check Out** buttons are disabled
- If you check out a prior revision of a package, but do not want to commit it as a new revision, right-click on the package and select **Package Control | Undo Check Out**

3.5.1.4.15.1 Review Package History - SCC Client

The File History command allows you to review the change history of version controlled packages. It also enables you to check out a prior revision of the package for editing, effectively allowing you to roll-back to a prior revision of the package.

The process for reviewing the package history of packages configured for version control with an SCC client (which includes products such as Visual Source Safe, TFS-SCC, ClearCase, Perforce, AccuRev, MKS Source Integrity and Serena Dimensions) differs from that for Subversion, CVS or TFS command line clients.

Use to:

- Review the change history of version controlled packages
- Revert to a previous revision of a version controlled package

How to:

Step	Action	See also
1	In the Project Browser, select the package for which to view change history.	
2	Right-click on the package and choose Package Control File History . The change history mechanism offered by the third party SCC provider displays.	
3	To import a prior revision of the package into your model, use the SCC History dialog to retrieve the revision, then close the dialog. The SCC client notifies Enterprise Architect that a different revision has been retrieved. Enterprise Architect then displays a prompt, asking whether to check-out the prior revision.	Example - Retrieve Prior Revision From SCC <small>[295]</small>
4	Optionally, click on either; <ul style="list-style-type: none"> • Yes, to check out the prior revision • No, to import the prior revision as read-only <p>If you click on Yes, the selected revision of the package file is retrieved from version control and imported into your model as a package that is checked out for editing. You can subsequently check in this revision as a new HEAD revision - effectively allowing you to revert a package to a prior revision.</p> <p>If you click on No, the selected revision of the package file is retrieved from version control and imported into your model, but the package remains flagged as checked-in, so it cannot be modified. Subsequently checking out the package updates the package to the latest revision before it is unlocked for editing.</p>	

Notes:

- If the selected package was already checked out in the current model, Enterprise Architect does not proceed with retrieving a prior revision
- If you check out a prior revision of a package, but do not want to commit it as a new revision, right-click on the package and select **Package Control | Undo Check Out**

Depending on your version control product, retrieving the prior revision can involve a number of prompts regarding overwriting the current local copy.

The following example details retrieval of a prior revision from a TFS-SCC version control configuration.

How to:

Step	Action	See also
1	Display the TFS File History dialog.	Package History - SCC <small>[294]</small>
2	Click on the Get button. The TFS-SCC client displays the Resolve Conflicts dialog. This dialog offers an option to Automerge All XML package files. DO NOT select this option. It is important to prevent any merging of Enterprise Architect's XML package files!	

3	Click on the Resolve button. The TFS-SCC client displays the Resolve writable file conflict dialog.	
4	Select the Overwrite local file/folder option. The existing working copy of the package file is overwritten by the prior revision retrieved from version control.	
5	Click the OK button. The TFS-SCC client redisplay the Resolve writable file conflict dialog; it should now show no conflicts.	
6	Click on the Close button. The TFS-SCC client redisplay the file History dialog.	
7	Click on the Close button. Enterprise Architect displays a prompt, asking whether to check-out the prior revision.	
8	Click on the: <ul style="list-style-type: none"> • Yes button to check-out the prior revision • No button to retrieve a read-only version of the package, that is NOT checked-out and is NOT editable. 	

3.5.1.4.16 Validate Package Configurations

You can test the validity of the version control settings associated with each version controlled package within your current model.

Access: **Project | Version Control | Validate Package Configurations**

How To:

Step	Action	See also
1	From the main menu choose Project Version Control Validate Package Configurations . The validation process scans the model database and verifies that the version control configuration associated with each version controlled package is fully specified in the current model. It also queries the corresponding version control provider to find the status of the package file associated with each version controlled package. The results of the validation process are sent to the Enterprise Architect Output window.	
2	Open the Version Control Settings dialog to complete the definition of any invalid or missing version control configurations.	Version Control Setup [268]
3	Click on an error message in the Output window to highlight the corresponding package in the Project Browser .	
4	Right-click a package node and choose Package Control Configure Package to open the Package Control Options dialog. Correct any problems with the version control details for the package.	Configure Controlled Package [282]

Step	Action	See also
	Correct any problems with the package's associated XML file.	

3.5.1.4.17 Resynchronize the Status of Version Controlled Packages

Enterprise Architect enables you to re-synchronize the version control status of version controlled packages within your project with the status reported by your version control provider. This can be useful if you are creating copies of your Enterprise Architect project, where checking in a package from one copy of the model leaves the package in the second copy of the model with an out of date version control status.

For a given package, the re-synchronization process queries the corresponding version control provider to find the status of the package file associated with the version-controlled package. If necessary, the process then updates the package flags within the model database, to synchronize the package status recorded in the model with the value reported by the version control provider.

Use to:

- Update the version control status of packages in your model, to align it with the status reported by their associated Version Control Configuration

How to:

Step	Action	See also
1	You can resynchronize either a single version controlled package or all version controlled packages within your project: Choose either; <ul style="list-style-type: none"> • Right-click the package in the Project Browser and select Package Control Re-synch Status With VC Provider • Project Version Control Re-synch Statuses of All Packages The results of the re-synchronization process are sent to the Enterprise Architect Output window.	
2	Double-click on any result message to select, in the Project Browser, the corresponding package.	

Notes:

- This process does not cause any package data to be either exported from your model to the associated package file, or imported from a package file into your model's package data
- If a package has been checked-out and modified with Enterprise Architect, but your version control provider reports the package file as checked-in, running this process marks the package within Enterprise Architect as being checked-in, without exporting and committing the pending changes. Subsequently checking-out the package imports the latest revision of the package file from version control, effectively discarding the uncommitted modifications from the model
- Similarly, if a package file is checked-out to you in your local working copy folder, but not in the Enterprise Architect model, running this process marks the package within the model as checked-out, but it does not import the associated package file from the version control system. Consequently, it is possible to check-in a package from Enterprise Architect that is potentially out of date, compared to the latest revision of the package file within the version control system

3.5.2 Tracking Changes

Enterprise Architect provides two separate but complementary facilities for tracking changes to data across the project.

Topics:

Topic	Detail	See also
Auditing of model changes	<p><i>Auditing</i> is a project-level feature, available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, that enables you to record model changes in Enterprise Architect</p> <p>By enabling this feature, model administrators can view a range of information regarding changes, such as:</p> <ul style="list-style-type: none"> • Who changed an element • How many elements they changed • When they changed the data • What the previous values were, and • What type of elements they changed 	Auditing ^[298]
Baselining and differencing to capture and roll back changes	<p>The Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions provide a facility to '<i>baseline</i>' or snapshot a model branch in XML format at a particular point in time, and store it within the model in compressed format</p> <p>More than one baseline can be stored against a single Enterprise Architect package; using baselines, you can compare packages at the current and earlier stages of development, using the <i>Compare</i> (Diff) utility</p> <p>The Compare utility is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect; it enables you to compare the current model with:</p> <ul style="list-style-type: none"> • A Baseline • An exported Enterprise Architect XML file on disk • A version-controlled Enterprise Architect XML file on disk 	<p>Package Baselines ^[308]</p> <p>The Compare Utility (Diff) ^[313]</p>

3.5.2.1 Auditing

Auditing is a project-level feature that enables model administrators to record model changes in Enterprise Architect.

Guide:

Topic	Detail	See also
Overview	<p>By enabling Auditing, you can view information on changes such as:</p> <ul style="list-style-type: none"> • Who changed an element 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • How many elements they changed • When they changed the data • What the previous values were, and • What type of elements they changed Auditing does not record changes to: <ul style="list-style-type: none"> • RTF Templates • Model Documents • Baselines • Profiles 	
Quickstart	You can quickly enable auditing and see it in action	Auditing Quickstart ^[307]
Settings	Once auditing is enabled within a project, you have a variety of options available for customizing what is recorded by the audit	Audit Settings ^[307]
The Audit View	To view what has been recorded by the audit, use the Audit View, which provides an interface to everything recorded by auditing If security is enabled, you must have Audit View permission to display data in the Audit View	Audit View ^[302] List of Available Permissions ^[205]
Model Views	You can also obtain a snapshot of selected items in the model, using the <i>Model View</i> facility In the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions of Enterprise Architect, this facility enables you to automatically generate a snapshot at intervals and, if there are changes in the items collected by the defined search, to trigger a notification to you of such changes This enables you to monitor work flow and other events of concern to you	Model View ^[464] Monitor Events ^[413]
RTF Report	You can generate an RTF report that includes the audit history information for the selected element or package, by choosing the <i>basic + audit</i> RTF template	RTF Dialog Options ^[1732]
Audit History	Using Auditing, you can track changes to an element or connector over time However, enabling Auditing also enables an Audit History tab in the System Output window, which summarizes all changes made to the selected element or connector	Audit History Tab ^[307]
Performance Issues	By enabling auditing on a project, you increase the time taken for most actions For most modeling tasks, this increase is insignificant, however, there are some instances where the difference is more substantial	Performance Issues ^[307]

Notes:

- The Auditing facility is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- **Warning** - If your site runs separate editions of Enterprise Architect, when Auditing is turned on in a project any Desktop or Professional edition users are locked out of the project; to restore access, turn Auditing off in the project from the Corporate, Business and Software Engineering, Systems Engineering or Ultimate edition instance of Enterprise Architect

3.5.2.1.1 Auditing Quickstart

You can quickly enable auditing and see it in action.

Access: **Project | Audit View: Audit Settings button**

How to:

Step	Action	See also
1	Select the Audit View menu option to open the Audit View	The Audit View ^[302]
2	Click on the Audit Settings button The Audit Settings dialog displays	Auditing Settings ^[300]
3	Select the Enable Auditing checkbox	
4	Click on the OK button to close the Audit Settings dialog	
5	Close the Audit View dialog	
6	Change and save your project; for example: <ul style="list-style-type: none"> • Add a new package • Add a new Class • Add a new connector • Change the name of an element • Delete an element 	
7	Open the Audit View again and click on the Refresh (or Load) button to display a record of the changes you made	

3.5.2.1.2 Auditing Settings

The Audit Settings dialog enables you to change what is recorded by the Auditing facility.

Access: **Project | Audit View: Audit Settings**

Use to:

- Define the areas of processing in Enterprise Architect to audit
- Administer your audit records
- Indicate the kind of model objects for which changes are to be recorded
- Configure auditing to record changes to only certain types of elements

Reference:

Action	Usage	See also
Enable Auditing	Turn the Auditing facility on	
Audit XMI Import	Include XMI importing in the audit As version control uses XMI, this option must be selected to record changes from checking out packages	Version Control ^[242]
Audit XMI Export	Include XMI exporting in the audit As version control uses XMI, this option must be selected to record changes from checking out packages	Version Control ^[242]
Audit Reverse Engineering	Include reverse engineering in the audit	
Use Database Timestamp	Use the database server's timestamp instead of each user's local timestamp; this improves security Not available for .eap project files	
Save Logs	Save a copy of the logged audit items currently held in the project; these items remain in the project - use Clear Logs to remove them Enterprise Architect prompts you to specify whether to save items covering a specific period of time <ul style="list-style-type: none"> Click on the No button to save all log items currently held in the database Click on the Yes button to display the Time Filter dialog, on which you select a standard time period or define your own This function can be accessed through the Automation Interface	Audit View Controls ^[304] Repository ^[1855]
Clear Logs	Permanently delete all log data from the current project; use the Save Logs button first, to save the audit records outside the project Enterprise Architect prompts you to specify whether to clear items covering a specific period of time <ul style="list-style-type: none"> Click on the No button to clear all log items currently held in the database Click on the Yes button to display the Time Filter dialog, on which you select a standard time period or define your own This function can be accessed through the Automation Interface	Audit View Controls ^[304] Repository ^[1855]
Load Logs	Load a previously saved set of logs back into the project; if the same record exists in both project and file, it is not reloaded This function can be accessed through the Automation Interface	Repository ^[1855]
Core	Record changes to elements (including attributes and operations), packages, connectors and some model-level information	

Action	Usage	See also
Standard	Record the same changes as the Core option, plus changes to diagrams	
Extended	Record the same changes as the Standard option, plus changes to security	
Maintenance	Audit maintenance elements only; that is: <ul style="list-style-type: none"> • Package • Requirement • Feature • Use Case • Actor • Note • Issue • Change 	
Core Structural	Audit maintenance elements plus some structural elements; that is: <ul style="list-style-type: none"> • Package • Class • Interface • Signal • Node • Component • Artifact • Part • Port • Device 	
All	Audit all elements	
Custom	Audit element types that you specify The Customize button is made available; click on this button to display a list of element types, and select the checkbox against each element type to include in the audit (or click on the Select All button to select every element type) Click on the OK button to save the selection	

Notes:

- As the number of records increases, the performance of the Audit View reduces; it is recommended that audit records that are not regularly required are saved to file, then cleared from the project, to help ensure high performance
- Connectors are audited when they are connected to an element that is included in the Audit Options
- If security is enabled, you must have **Audit View** permission to turn Auditing on, and **Audit Settings** permission to change Audit settings

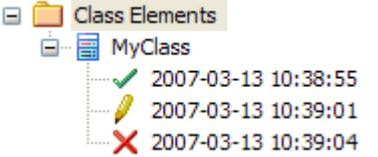
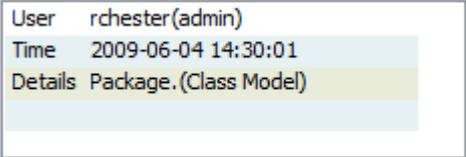
3.5.2.1.3 The Audit View

The Audit View provides an interface to the information that has been recorded by auditing.

Access: [Project | Audit View](#)

Guide:

The Audit View is divided into three main areas:

Action	Usage	See also
View controls	The view controls provide a variety of settings for controlling auditing and the display of audit records	Audit View Controls ^[304]
Audit tree	<p>The audit tree displays the log items that have been recorded by auditing</p> <p>What is displayed in the tree is affected by the view controls, such as:</p> <ul style="list-style-type: none"> • Sorting • Filter (by time) • Mode • Auditing settings (what was actually recorded) <p>If synchronizing with the Project Browser, it is also affected by the package, diagram or element you have selected</p>  <p>In the audit tree:</p> <ul style="list-style-type: none"> • The green tick indicates a creation • The yellow pencil indicates an edit • The red cross indicates a deletion <p>Right-clicking an element in the audit tree (such as <i>MyClass</i>) displays a context menu that enables you to locate the selected element in:</p> <ul style="list-style-type: none"> • The Project Browser • Any diagrams in which it exists 	Audit View Controls ^[304] Auditing settings ^[304] The Project Browser ^[441]
Record display	<p>The record display is in two parts: the identity of the selected change, and the actual change made</p> <p>The data in the record display is determined by the view controls and mode and, if synchronizing with the Project Browser, by the package, diagram or element you have selected</p> <p>Identity:</p>  <p>The identity of a change consists of:</p>	Audit View Controls ^[304] The Project Browser ^[441]

Action	Usage	See also
	<ul style="list-style-type: none"> The Windows username of the user that made the change; if security is enabled, the name is of the format <i>WindowsUsername(SecurityUser)</i> When the change was made The path of the change; for example: Class <i>Class1</i> - Attribute <i>Att1</i> - Attribute Constraint <i>Constraint</i> <p>Changes:</p> <p>Changes are displayed in a table format, showing the:</p> <ul style="list-style-type: none"> Property (or data item) name Its original value before the change and Its value after the change <p>If you double-click on an item in the list of changes (or right-click and select the Show Differences context menu option) the Difference window displays:</p>  <p>This shows the specific changes that have been made, highlighting the edited, created, deleted or formatted characters</p> <p>Changes to the <i>format</i> of text in the element, made through the element Properties dialog, are not apparent in the initial table; they are visible in the Difference window, identified by HTML formatting tags</p>	

Notes:

- If security is enabled, you must have **Audit View** permission to display data in the Audit View

Learn More:

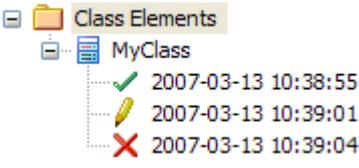
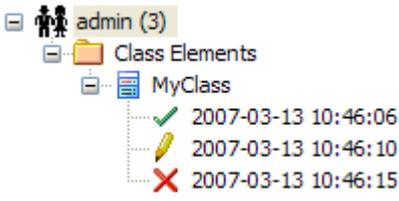
- [List of Permissions](#) ^[208]

3.5.2.1.3.1 Audit View Controls

The **Audit View** controls provide a variety of settings for controlling auditing and the display of audit records.

Reference:

Action	Usage	See also
Load or Refresh	Click to reload the Audit Tree, updated with any new audit results	
Search	Click to search through log items for a particular area - you can search by Name, Type or GUID The items are loaded and filtered with the current Sort By, Time Filter and Mode settings	

Action	Usage	See also
	If you refresh the Audit View, you must run the search again	
Audit Settings	Click to open the Audit Settings dialog	Audit Settings 
Sort-by	<p>Select the appropriate radio button for the required display setting:</p> <ul style="list-style-type: none"> • Type - changes are grouped under element type (such as Class or Requirement), and then grouped under the changed element  <ul style="list-style-type: none"> • User - changes are grouped under user name, each with the number of changes for that user <p>Under each user name, changes are grouped as for the Type sort</p> 	
Filter By Date/Time	<p>Select to enable the Filter Settings button, to filter the audit results by time period</p> <p>Changes that occur <i>outside</i> the selected filter period are not shown in the Audit View</p>	
Filter Settings	<p>Click to display the Time Filter dialog, to set the filter time period:</p> <ul style="list-style-type: none"> • Today - to display changes occurring today • Previous Hour - to display changes occurring in the last 60 minutes • Previous 24 Hours - to display changes occurring in the last 24 hours • Previous Week - to display changes occurring in the last 7 days • Previous 30 Days - to display changes occurring in the last 30 days • Previous Year - to display changes occurring in the last 365 days • Custom - to define your own time period, using the From and To fields <p>The six pre-configured time periods automatically update when you click on the Refresh button; custom periods are static and do not automatically update</p>	

Action	Usage	See also
	<p>If you have set a filter period, and you deselect the Filter By Date/Time checkbox, the period remains set; the custom time period, too, is retained so that you can re-use it or modify it later if required</p>	
Status Text	<p>Read to see which:</p> <ul style="list-style-type: none"> • Mode has been selected and • Time filter is being applied to the data 	
Mode	<p>Click to display a menu of options to change the mode of the Audit View; select:</p> <ul style="list-style-type: none"> • Standard - to automatically synchronize with the Project Browser; where changes have been made, the Audit View reflects your selection from the Project Browser - if you click on: <ul style="list-style-type: none"> • An element, the Audit View displays the history for that element • A package, the Audit View displays the history for that package and its immediate children (but not the contents of nested packages) • A diagram, the Audit View displays the history for that diagram and its contents (which could be drawn from a wide area of the Project Browser) • Advanced - to load large sets of log items independent of the Project Browser; in this mode, a special Audit Settings group can be displayed in the Audit Tree, which records: <ul style="list-style-type: none"> • When Auditing has been enabled and disabled • Who made the change • The date and time of the change • Changes to the Audit Settings • When Audit Operations are executed • Security changes (which can be browsed in the same way as other changes) • Deleted - to display only deleted records, but otherwise data is shown as in Advanced mode; records can be sorted by element type or by user as required • Raw - to display all audit records in chronological order without sorting, enabling you to see a progression of changes; this can be especially useful in determining date-time inconsistencies <p>Any search and filtering you define still apply, enabling you to view all of today's changes in order, or all changes for a particular element in order, or both</p> <p>Additional database information is displayed; this additional information might be insignificant or only in machine-readable format</p>	<p>The Audit View ³⁰²</p>

3.5.2.1.4 Audit History Tab

When Auditing is turned on, an Audit History tab is enabled in the System Output window.

Access: [View | System Output > Audit History tab](#)

Guide:

Topic	Detail	See also
Prerequisite	To see this tab, you must have the Audit View open The tab continues to display if you subsequently close the Audit View	The Audit View ^[302]
Information Shown	The tab shows a history of changes to whichever element or connector you have selected in the: <ul style="list-style-type: none"> • current diagram • Project Browser • Audit View, or • Package Browser As you select different elements or connectors, the Audit History tab automatically updates to reflect your current selection For each change made to the element or connector, the tab shows: <ul style="list-style-type: none"> • Who made the change • When the change was made • Where the change was made • The value of the characteristic before the change • The value of the characteristic after the change 	

Notes:

- If security is enabled, you must have **Audit View** permission to display data on the Audit History tab

Learn More:

- [System Output Window](#)^[127]
- [List of Permissions](#)^[208]

3.5.2.1.5 Auditing Performance Issues

Enabling auditing on a project increases the time taken for most actions.

For most modeling tasks, this increase is insignificant; however, there are some situations where the difference is more substantial.

Topic	Detail	See also
Large Deletions	Deleting large packages or package structures, or large numbers of elements, takes significantly longer with auditing on You might disable auditing before performing such a deletion	Auditing Settings [300]
XMI Imports	Importing XMI takes longer with auditing enabled A project-level option is provided for disabling auditing of XMI Imports	Import From XMI [323] Auditing Settings [300]
Reverse Engineering	Reverse engineering code takes longer with auditing enabled A project-level option is provided for disabling auditing of reverse engineering	Import Source Code [1508] Auditing Settings [300]
Replication	You cannot remove replication from a model with Auditing enabled If you have to remove replication, disable Auditing and - if prompted to do so - allow Enterprise Architect to roll back the database version; you can then remove replication	Remove Replication [192] Auditing Settings [300]

3.5.2.1.6 Audit View Performance Issues

Most operations in the Audit View are affected by the volume of use of the database - both by other facilities and by auditing itself.

Some potential problems and their solutions are outlined below:

Topic	Detail	See also
Navigating the Project Browser Within Auditing is Slow	Try setting the time filter to a period in the immediate past, such as Today , Previous 24 Hours or Previous Week ; this time period updates each time you open or refresh the Audit View	Audit View Controls [304] Auditing Settings [300]
The Audit View is Slow in Loading and Changing Modes	Save log items outside the project with the Save Logs button; if you then clear the logs you have just saved, the load time of the Audit View is reduced You can reload logs into the project at any time, using the Load Logs button	
Navigating the Audit Tree is Slow	Close the Audit History tab in the Output window	Audit History Tab [307]

3.5.2.2 Package Baselines

Enterprise Architect includes tools to help you manage and review changes to your models over time. These tools apply the concepts of *Baselines*, *Differencing* and *Merges*.

You use Baselines, Differencing and Merges essentially to compare two snapshots of a specific part of your project, to capture the differences between them and either roll back or incorporate selected changes or all changes.

Topic	Detail	See also
Baselines	<p>Enterprise Architect provides a facility to create a Baseline or 'snapshot' of the contents of a selected package and its child packages at a particular point in time; this enables you to later compare that branch of the model at that time with the current state of the branch</p> <p>Baselines are stored in the same XML format as is used for version control, but are stored within the project in compressed format</p> <p>You can also have parallel copies of parts of your model for team development, and create Baselines within each copy to merge changes into the project master</p>	<p>Baselines ^[310]</p>
Differencing	<p>Differencing (<i>Diff</i>, or <i>Compare</i>) enables you to explore the differences between:</p> <ul style="list-style-type: none"> • The current state of a specific part of your project, and • Previous or parallel versions captured in a Baseline or an XML 1.1 file on disk <p>The section includes an example of such a comparison</p>	<p>The Compare Utility (Diff) ^[313]</p> <p>Example Comparison ^[315]</p>
Merges	<p>Once Differencing is complete, you can merge information from the Baseline into the current project; it is not possible to go the other way</p> <p>You can:</p> <ul style="list-style-type: none"> • Merge information manually, change by change • Merge information automatically by electing to merge in all changes in one batch procedure • Revert completely to the original Baseline by importing the stored XML directly • Merge information and elements from a Baseline in a different project, making it possible to keep multiple versions of a single model in synch <p>The merge options are available through the toolbar, context menus and the keyboard on the Compare Utility tab, which shows the results of a comparison</p>	<p>Compare Utility Tab Options ^[316]</p>

Notes:

- Package Baseline facilities are available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- The Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions provide another facility, Auditing, which you can switch on to perform continuous monitoring of changes across the project; you can dovetail your use of each facility to meet the range of your change management requirements
- If a package under version control forms part of a Baseline, and that package is checked in to the model, you cannot merge the original data from the Baseline into that package
- You can also obtain a snapshot of selected items in the model, using the Model Views facility; this facility enables you to automatically generate the snapshot at intervals and, if there are changes in the items collected by the defined search, to trigger a notification to you of such changes, which enables you to monitor work flow and other events of concern to you
- If security is enabled you must have **Baselines - Manage** permission to create, import and delete Baselines, and **Baselines - Restore** permission to merge data from a Baseline; security permissions

are not required to select an existing Baseline and perform a comparison with the model

Learn More:

- [Auditing](#) ^[298]
- [List of Permissions](#) ^[205]
- [Model Views](#) ^[464]
- [Monitor Change Events](#) ^[413]

3.5.2.2.1 Baselines

Enterprise Architect provides a facility to 'Baseline' (snapshot) a model branch at a particular point in time for later comparison with the current package state.

Topic	Detail	See also
Baselines	<p>Baseline comparison is most useful for determining the changes made to the model during development compared to some Baseline saved at a crucial point - for example the completion of a phase or version iteration</p> <p>More than one Baseline can be stored against a single Enterprise Architect package</p> <p>Baselines are particularly useful during requirements management to check for changes, additions and deletions that have occurred since the start of the current work phase; knowing how a model has changed is an important part of managing change and the overall development process</p> <p>Baselines are stored within the model in compressed XML format; you can save a Baseline to an external XML file for storage or archive, or for distributing to other users working on models derived from a master project</p> <p>Baselines are generally used in conjunction with the <i>Compare</i> utility</p>	<p>The Compare Utility (Diff) ^[313]</p>
Scenario	<p>A typical scenario for using Baselines would be:</p> <ul style="list-style-type: none"> • Create the base model branch to a sufficient point to create a Baseline (checkpoint); create and store the Baseline as Version 0.1a • As work continues on development, managers and developers can check the current model branch against the Baseline for important modifications, additions and deletions <p>The Compare utility can be invoked from the Baseline dialog to check the current model branch against the stored version</p> <ul style="list-style-type: none"> • As required, minor Baselines can be created to check recent progress; these 'temporary Baselines' are useful for managing change when a lot of work is being done and it is important to only see what has changed in, for example, the last 24 hours • At sign-off or the move to a new version/phase, a major Baseline can be created to capture the new state of the model 	<p>Manage Baselines ^[311]</p> <p>Create Baselines ^[312]</p>

Topic	Detail	See also
	Minor Baselines created earlier can be deleted if required to save space	
Considerations	<ul style="list-style-type: none"> Baselines are based on the GUID or unique ID of a particular package: <ul style="list-style-type: none"> Enterprise Architect checks for that ID as the root element within the XML document being used as a Baseline When you export a package to XML, the package you export is the root element; likewise when you create a Baseline, the current package is the root package of the XML Baseline When you save information in a version control system, the current version-controlled package is again the root package of the document It is not useful to create a Baseline by importing an XML package file created by version controlling a package that itself contains version-controlled child packages; that type of XML package file contains stubs for the child packages, not full information on the child packages and elements If a package under version control forms part of a Baseline, and that package is checked in to the model, you cannot merge the original data from the Baseline into that package XML files must be in the same format used by the Baseline engine - currently the UML 1.3 XML 1.1 format (plus Enterprise Architect extensions), which contains all the information necessary to reconstruct a UML model, even a UML 2.x model 	

Notes:

- The Baseline facility is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- The Compare utility is available in the *Professional* edition of Enterprise Architect, as well as in the Corporate and extended editions above

3.5.2.2.1 Manage Baselines

Enterprise Architect provides a range of facilities for working with and managing Baselines, through the Manage Baselines dialog.

Access: [Right-click model branch package | Package Control | Manage Baselines](#)

Field	Usage	See also
Current Baselines For Package: <Name>	<p>Review the baselines for the current model branch, listed by version reference with the highest alphabetical/numerical value at the top</p> <p>If an entry is longer than the display area, a horizontal scroll bar displays at the bottom of the panel; use this to scroll to the text that is not shown</p>	Create Baselines 312

Field	Usage	See also
Show Differences	Run the Compare ^[313] utility on the selected Baseline and the current model branch, to display the differences ^[315] between the two	The Compare Utility ^[313] Example Comparison ^[315]
Restore to Baseline	Completely restore the model branch from the selected Baseline	
New Baseline	Create a new Baseline	Create Baselines ^[312]
Delete Selected	Delete the selected Baseline	
Load Other Baselines	<p>Display a drop-down menu that enables you to load Baselines from another model, in either a .EAP file or a DBMS file</p> <ul style="list-style-type: none"> For .EAP files, a browser displays; locate the required project file For DBMS files, the Windows Data Link Properties dialog displays; select the data provider and click on the OK button to display the Select Data Source dialog, from which you select the required project <p>In either case, the <i>Connected To:</i> message at the bottom of the Manage Baselines dialog changes to the name of the alternative model</p> <p>To return the dialog to the original project, select the third option on the drop down list: Load From Selected Package</p>	
Import File	Import an XML 1.1 file from the file system as a new Baseline for this current model branch	
Export File	Export the selected Baseline to an XML file on disk	
Compare Model to File	Compare the selected model branch with an XML 1.1 file on disk; a browser displays, which you use to locate the file	
Options	Set filters ^[314] to make the comparison more specific	Compare Options ^[314]

3.5.2.2.1.2 Create Baselines

Access: Right-click model branch package | Package Control | Manage Baselines: New Baseline button

The New Baseline dialog displays

Field	Usage	See also
Name	Display the package name of the currently selected model branch	Manage Baselines ^[311]
Version	<p>Type a unique version reference for this Baseline - this can consist of any alphanumeric characters</p> <p>The Manage Baselines dialog sorts the Baselines according to the value of this field</p>	
Include	Include the entire sub-package hierarchy of this branch in the Baseline -	

Field	Usage	See also
Sub-packages	defaults to selected If you deselect the checkbox, only the immediate contents (XML stubs) of the package are included in the Baseline	
Note	Edit the default current time and date to any other value The field is a single-line entry, for display on the Manage Baselines dialog (a one-line-per-entry list)	
OK	Click to create a new Baseline and return to the Manage Baselines dialog	

3.5.2.2.1.3 The Compare Utility (Diff)

Enterprise Architect has a comprehensive and powerful built in *Compare (diff)* utility, which enables you to:

- Explore what has changed within a model over time
- Explore how previous versions of a model branch differ from what is currently in the model
- Perform a full model comparison by exporting all of Model A to XML, then using **Compare Model to File** from within the current model (Model B)

Comparing and checking model development at various points in the process is an important aspect of managing change and development, monitoring what is being modified and ensuring the development and design process is on track.

Access:

- **Project Browser package context menu | Package Control | Manage Baselines: Show Differences** - use the Baseline dialog
- **Project Browser package context menu | Package Control | Compare with XML File** (package not under version control)
- **Project Browser package context menu | Package Control | Compare with Controlled Version** (package under version control)

Use to:

- Compare a model branch in Enterprise Architect with a Baseline created using the Baseline functionality (Corporate, Business and Software Engineering, System Engineering and Ultimate editions)
- Compare a model branch in Enterprise Architect with a Baseline stored in a different model
- Compare a model branch in Enterprise Architect with an XML 1.1 file on disk created previously using the Enterprise Architect XML export facility (user selects file)
- Compare a model branch in Enterprise Architect with the current version-controlled XML 1.1 file on disk as created when using Version Control in Enterprise Architect (file automatically selected)

Guide:

Topic	Detail	See also
Differencing With Baselines	As a Baseline is stored within a model and contains all the information, elements and connections for a package at a point in time, it can be used within Enterprise Architect to track changes to model elements over time The Differencing engine first builds a representation of the	Compare Options ^[314] Create Baselines ^[312] Example Comparison ^[315]

Topic	Detail	See also
	<p>current package in memory, based on what is currently in the model</p> <p>It then compares this with the stored Baseline, highlighting changes, new elements, missing elements and elements that have been moved to other packages</p> <p>It is possible to filter the resultant output to display only one particular kind of change: for example, additions to the model.</p> <p>If a Baseline has been created to ignore child package content, a comparison between that Baseline and the model does not include any child package content in the model</p> <p>See the example model comparison</p>	

Notes:

- This utility is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- You cannot compare the current model with an XML 2.1 file; the utility can only compare with an XML 1.1 file

3.5.2.2.1.4 Compare Options

The Compare Options dialog enables you to refine the output of the Compare utility when it compares the current model with a Baseline.

To display the dialog, either:

- Click on the **Options** button on the Manage Baselines dialog
- Click on the **Compare Options** icon on the Compare Utility tab toolbar

Field	Usage	See also
Always Expand to Differences	<p>Always display the list of elements fully expanded to show changes</p> <p>If you deselect the checkbox, when the Compare Utility tab is first opened it lists the package contents to element level, and you expand each element as required to show the changed items</p> <p>For large branches of the model, it is better to leave the checkbox unselected</p>	
Show Elements that are	<p>List elements that:</p> <ul style="list-style-type: none"> • Have been changed since the Baseline was created • Are in the Baseline only (that is, have been deleted from the model since the Baseline was created) • Are in the model only (that is, have been created since the Baseline was created) • Have not changed since the Baseline was created (you might generally leave this checkbox unselected) 	
Suppress these Changes	<p>Exclude:</p> <ul style="list-style-type: none"> • Changes to diagrams • Changes to the Date Modified field for an item • Changes to the Date Created field for an item 	

Field	Usage	See also
	<ul style="list-style-type: none"> • Child items of a deleted item • Changes to advanced properties (defaults to selected) 	

If the Compare Utility tab shows the results of a Baseline comparison, when you click on the **OK** button the display refreshes to refine the information according to the options you have selected.

Learn More:

- [Compare Utility Tab Options](#) ³¹⁶

3.5.2.2 Example Comparison

The diagram below shows the result of a comparison between a package (Business Process Model) in the current project and that package in a Baseline (version 0.1) captured at an earlier date.

The results of the comparison are displayed on the Compare Utility tab.

Model Elements	Status	Property	Model	Baseline
Business Process Model		Abstract	false	false
Rules		Alias		
People must wear safety belts in all se...	Changed	Author	Suzanne Pearson	Suzanne Pearson
No more than 5 persons in a vehicle	Changed	Date Created	16/01/2009 12:07:15 PM	16/01/2009 12:07:15 PM
Passengers under 8 years old must b...	Model only	Date Modified	5/06/2009 10:15:00 AM	16/01/2009 12:07:30 PM
Customer		Complexity	1	1
RuleFlow		Filename	C:\Documents and Settings\...	C:\Documents and Settings\...
Eligibility		Language	C#	C#
address	Changed	IsLeaf	false	false
Application	Moved	IsSpec	false	false
Status	Baseline only	IsRoot	false	false
Constraints		Keywords		
AllowableValues{Accept, Reje...	Baseline only	Multiplicity		
		Name	Application	Application
		Notes		
		Parent Package	Business Rules Model	Business Process Model
		Persistence		
		Phase	1.0	1.0

Guide:

Topic	Detail	See also
Interpretation	<p>A hierarchy of model elements is displayed in the left-hand pane</p> <p>It is clearly visible, from the Status column and from the triangle-based icons, which items in the hierarchy have, since the Baseline was captured, been:</p> <ul style="list-style-type: none"> • Changed (<i>Changed</i>) • Deleted from the model (<i>Baseline only</i>) • Added to the model (<i>Model only</i>) or • Switched to different packages (<i>Moved</i>) <p>If you click on an item with a Status entry in the left hand pane, the right-hand pane displays a table of properties showing the values of those properties in the current model and in the Baseline</p>	<p>Compare Utility Tab Options ³¹⁶</p>

Topic	Detail	See also
	<p>For each property where there is a difference between the model and the Baseline, the row is highlighted; for example, the Class element named <i>Application</i> was moved from the <i>Business Process Model</i> package (as shown in the Baseline column) to the <i>Business Rules Model</i> package on 5 June 2009 (as shown in the Model column)</p> <p>The Compare Utility tab enables you to perform operations (such as merging or rolling back changes) on the reported information, using the toolbar, context menu and keyboard</p>	
Increase Level of Detail	<p>The right panel of the Compare Utility tab might, for some fields, display only part of the value (such as <i>Advanced Properties</i>, above)</p> <p>It might also not be immediately obvious what a change is</p> <p>In either case, you can double-click on the property to display full details and to highlight the exact differences; the following example shows the highlighted changes to the Parent Package property</p> 	

3.5.2.2.1 Compare Utility Tab Options

The Compare Utility tab enables you to perform operations on the reported information, using the toolbar, context menu, Merge dialog and certain keyboard keys.

- The toolbar is at the top of the left-hand panel; the icons operate either on the comparison as a whole or on the currently-selected item in the left hand panel of the Compare Utility tab
- Each item in the hierarchy has a context menu, which you display by right-clicking on the item; the options displayed depend on the level of the item in the hierarchy
- The Merge dialog enables you to specify which changes to roll back in the model from the baseline
- You can use a selection of keyboard keys to move up and down the hierarchy, or to roll back changes

Reference - Toolbar:



Action	Usage	Shortcut	See also
Refresh	Re-run the comparison to refresh the current display		
Merge To Model	Merge the values of the currently-selected item in the Baseline back into the model		
Next Change	Highlight the next changed item (skips <i>Moved</i> items)		
Previous Change	Highlight the previously-changed item		
Expand All	Fully expand the selected item		
Collapse All	Collapse the changed items in the selected item		

Action	Usage	Shortcut	See also
Expand To Changed Items	Expand the selected item to show changed items only (in the event that you have selected to also show [314] unchanged items in the comparison)		Compare options [314]
Find in Project Browser	Highlight the item in the Project Browser		
Log To XML	Log the changes to an XML file A browser displays, on which you specify the file name and location		
Compare Options	Display the Compare Options dialog		Compare Options [314]
Manage Baselines	Display the Manage Baselines [412] dialog		Manage Baselines [314]
Help	Display the Help topic Package Baselines [308]		Package Baselines [308]

Reference - Context Menu:

Action	Usage	Shortcut	See also
Merge from Baseline Add from Baseline	Restore the item in the model to the Baseline state, or restore a deleted item from the Baseline		
Delete from Model	Remove a recently-created item from the model		
Merge From Baseline (with Options)	(For the root node of the hierarchy on the Compare Utility tab) Display the Merge dialog (see below) which enables you to specify options for rolling back the whole model branch to the Baseline state		
Refresh	(Object-level items) Re-run the comparison to refresh the current display		
Find in Project Browser	Locate and highlight the item in the Project Browser		
Expand All	Fully expand the selected item		
Expand To Changed Items	Expand the selected item to show changed items only		
Collapse All	Collapse the changed items in the selected item		
Log To XML	Log the changes to an XML file A browser displays, on which you specify the file name and location.		
Compare Options	Display the Compare Options dialog		Compare Options [314]

Reference - Merge Dialog:

Field	Usage	Shortcut	See also
Changed	Restore all changed items in the model branch to the Baseline state		
In Baseline Only	Restore all deleted items to the model branch from the Baseline		
In Model Only	Remove all recently-created items from the model branch		
Moved	Restore all moved items to their original locations, as identified in the Baseline		
Full Restore from XMI	Completely restore the model branch to the version held in the Baseline XMI 1.1 file, (using the XMI Import function). (Automatically selects all the other options.)		

Reference - Keyboard:

- (**Ctrl+↓**) - expand and highlight the next changed item
- (**Ctrl+↑**) - expand and highlight the previous changed item
- (**Ctrl+←**) - undo the changes for a selected item (roll back to the Baseline values)

3.5.3 Model Transfer

Enterprise Architect enables you to transfer data between projects, using three different mechanisms.

Topics:

Topic	Detail	See also
XMI Import and Export	Packages can be exported from and imported into Enterprise Architect models This greatly improves the flexibility and robustness of Enterprise Architect models by enabling Analysts and Modelers to externalize model elements in XMI for version control, distributed development, post processing and transferring packages between models	XMI Import and Export ^[319]
CSV Import and Export	You can import and export information about Enterprise Architect elements in CSV format This enables you to store and retrieve information from spreadsheet tools such as Microsoft Excel	CSV Import and Export ^[337]
Data Transfer	Enterprise Architect enables you to move complete models between files, and between files and repositories, row by row, table by table	Perform a Project Data Transfer ^[343]

3.5.3.1 XMI Import and Export

XML Metadata Interchange (XMI), defined by the OMG and based on XML, is an open standard file format that enables the interchange of model information between models and tools.

Topics:

Topic	Detail	See also
<p>XMI and Enterprise Architect</p>	<p>Enterprise Architect uses XMI as a method of importing and exporting model specifications between different UML packages, Enterprise Architect projects and other tools that support XMI</p> <p>Enterprise Architect supports the XMI 1.1, 1.2 and 2.1 specifications. XMI 1.1 has support for UML 1.3, whereas XMI 2.1 has support for UML 2.x.</p> <p>Enterprise Architect does not fully support the older 1.0 specification. When importing or exporting to XMI 1.0, some loss of data occurs due to the limitations of XMI 1.0</p> <p>With XMI, model details can be exchanged between different UML tools and other tools that are capable of using XMI</p> <p>Limited support for export to Rational Rose is provided using the Rose version of the XMI 1.1 specification, as implemented by Unisys for Rational products.</p> <p>Packages can be exported from and imported into Enterprise Architect models, which greatly improves the flexibility and robustness of the models by enabling Analysts and Modelers to externalize model elements in XMI for:</p> <ul style="list-style-type: none"> • Version control • Distributed development • Post processing and • Transferring packages between models <p>When performing Enterprise Architect-to-Enterprise Architect transfers, ensure that either XMI version 1.1 or 2.1 is selected.</p> <p>When importing an XML file over an existing package, all information in the current package is deleted first, before importing from the XML file. Please make sure you do not have important changes that you do not want to lose before importing from XML.</p>	
<p>XMI Tasks</p>	<p>Tasks you might perform in importing and exporting XMI include:</p> <ul style="list-style-type: none"> • Setting XML Options - XMI import, export and package control all rely on saving and loading XML files; you can set a number of options to streamline this process • Exporting a package to XMI in XMI 2.1 (and earlier) • Importing from XMI with support for XMI 2.1 (and earlier) 	<p>XML Specifications ^[436]</p> <p>Export to XMI ^[320]</p> <p>Publish Model Package ^[321]</p> <p>Import from XMI ^[323]</p> <p>Controlled Packages ^[327]</p> <p>Manual Version Control with XMI ^[334]</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> Setting up controlled packages Manually controlling a package by linking it to an XMI file Batch exporting controlled packages Batch importing controlled packages Factoring in the limitations of XMI Applying a UML Data Type Definition (DTD) 	Batch XML Export ^[332] Batch XML Import ^[333] Limitations of XML ^[325] The UML DTD ^[326]

Notes:

- XML 2.1 exported by Enterprise Architect 7.0 (or later) might not be correctly imported into earlier versions of Enterprise Architect.
- When performing Enterprise Architect-to-Enterprise Architect transfers, ensure that either XML version 1.1 or 2.1 is selected.

Learn More:

- [XML/XML Technology](#)

3.5.3.1.1 Export to XMI

The Export Package to XMI dialog is used to export a package to various XMI (XML-based) formats. This dialog, by default, allows exporting to either **XMI 1.1** or **XMI 2.1**. The exported XMI file contains all the information required for completely round-tripping the package within Enterprise Architect; that is, performing Enterprise Architect-to-Enterprise Architect transfers.

Access: Click on a package in Project Browser, then:

Project | Model Import/Export | Export Package to XMI File
Right-click | Import/Export | Export Package to XMI

Use to:

- Move Enterprise Architect model elements between models, for distributed development, manual version control and other benefits

Reference:

Field/Option/Button	Usage	See also								
Root Package	Indicates the name of the selected package									
Filename	Specify the file path where the XMI file is to be generated									
Export Type	<table border="1"> <thead> <tr> <th>Option/Button</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>XMI 1.1</td> <td>Export the selected package to XMI 1.1</td> </tr> <tr> <td>XMI 2.1</td> <td>Export the selected package to XMI 2.1</td> </tr> <tr> <td>Publish</td> <td>Opens the Publish Model Package dialog for exporting to other versions of XML</td> </tr> </tbody> </table>	Option/Button	Usage	XMI 1.1	Export the selected package to XMI 1.1	XMI 2.1	Export the selected package to XMI 2.1	Publish	Opens the Publish Model Package dialog for exporting to other versions of XML	Publish Model Package ^[321]
	Option/Button	Usage								
	XMI 1.1	Export the selected package to XMI 1.1								
	XMI 2.1	Export the selected package to XMI 2.1								
Publish	Opens the Publish Model Package dialog for exporting to other versions of XML									

View XMI	View the exported XMI file	
Export	Initiate XMI Export	
Close	Close this dialog	
Help	Display this Help topic	
Progress	Lists the progress of the XMI Export	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XMI** permission to export to XMI
- Alternate Images used in the package being exported are not included in the exported XMI file unless the **Export Alternate Images** option is selected in the XML Specifications section of the Options dialog
- The Log file for the package being exported is not generated unless the **Write Log** option is selected in the XML Specifications section of the Options dialog

Learn More:

- [Limitations of XMI](#)^[325]
- [List of Available Permissions](#)^[205]
- [Manual Version Control with XMI](#)^[334]
- [XML Specifications](#)^[436]

3.5.3.1.2 Publish Model Package

The Publish Model Package facility is used to export a package to various XML formats, including XMI 1.1, XMI 2.1 and BPMN 2.0 XML.

Access: Click on a package in **Project Browser**, then:

Project | Model Publisher
Project | Model Import/Export | Export Package to XMI File: Publish
Right-click | Import/Export | Export Package to XMI: Publish

Use to:

- Move Enterprise Architect model elements between models, for distributed development, manual version control and other benefits
- Export BPMN 2.0 models to BPMN 2.0 XML
- Export Enterprise Architect model elements to Rational Rose and other tools that implement the:
 - UML 1.3 XMI 1.1 / XMI 1.0 standard
 - UML 1.4 XMI 1.2 standard, or
 - UML 2.1 XMI 2.1 standard

Reference:

Field/Option/ Button	Usage	See also
Root Package	Indicates the name of the selected package	
Filename	Specify the file path where the XML file is to be generated	

XML Type	Select the XML/XMI version to export the package to																			
General Options	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Export Diagrams</td> <td>Export all the diagrams in the selected package</td> </tr> <tr> <td>Export Alternate Images</td> <td>Export the alternate images used in the diagram package</td> </tr> <tr> <td>Format XML Output</td> <td>Format the output into readable XML (this takes a few seconds at the end of the run)</td> </tr> <tr> <td>Write Log File</td> <td>Write a log of the export activity (recommended) The log file is saved to the directory into which the package is exported</td> </tr> <tr> <td>Use DTD</td> <td>Use the XMI DTD when exporting to XMI 1.0 Setting this option validates the correctness of the XML; it performs syntax checks that no syntactical errors have occurred</td> </tr> <tr> <td>Exclude EA Tagged Values</td> <td>Exclude Enterprise Architect-specific information from the XML output</td> </tr> <tr> <td>Unisys/Rose Format</td> <td>Export in Unisys/Rose compatible XMI format</td> </tr> <tr> <td>Generate Diagram Images</td> <td>Generate the exported diagrams in the specified format The generated diagrams are saved in a separate directory into which the XML file is exported</td> </tr> </tbody> </table>	Field	Usage	Export Diagrams	Export all the diagrams in the selected package	Export Alternate Images	Export the alternate images used in the diagram package	Format XML Output	Format the output into readable XML (this takes a few seconds at the end of the run)	Write Log File	Write a log of the export activity (recommended) The log file is saved to the directory into which the package is exported	Use DTD	Use the XMI DTD when exporting to XMI 1.0 Setting this option validates the correctness of the XML; it performs syntax checks that no syntactical errors have occurred	Exclude EA Tagged Values	Exclude Enterprise Architect-specific information from the XML output	Unisys/Rose Format	Export in Unisys/Rose compatible XMI format	Generate Diagram Images	Generate the exported diagrams in the specified format The generated diagrams are saved in a separate directory into which the XML file is exported	Image Manager ^[593]
	Field	Usage																		
	Export Diagrams	Export all the diagrams in the selected package																		
	Export Alternate Images	Export the alternate images used in the diagram package																		
	Format XML Output	Format the output into readable XML (this takes a few seconds at the end of the run)																		
	Write Log File	Write a log of the export activity (recommended) The log file is saved to the directory into which the package is exported																		
	Use DTD	Use the XMI DTD when exporting to XMI 1.0 Setting this option validates the correctness of the XML; it performs syntax checks that no syntactical errors have occurred																		
	Exclude EA Tagged Values	Exclude Enterprise Architect-specific information from the XML output																		
	Unisys/Rose Format	Export in Unisys/Rose compatible XMI format																		
Generate Diagram Images	Generate the exported diagrams in the specified format The generated diagrams are saved in a separate directory into which the XML file is exported																			
Some of the above options might be unavailable, based on the selected XML Type																				
Stylesheet	Select an XSL Stylesheet to post-process the XML content before saving to file (<i>Optional</i>) You can post-process the XML content before saving the package to file, using an XSLT to convert the output to HTML, XSL, code or other versions of XML; to do this, you must import the required style sheet into the project through the Resources window	Resource Window ^[789]																		
View XML	View the exported XML file																			
Export	Initiate XML Export																			
Close	Close this dialog																			
Help	Display this Help topic																			
Progress	Lists the progress of the XML Export																			

Notes:

- When exporting and importing with XMI 1.0 with Enterprise Architect, some loss of data occurs due to the limitations of XMI 1.0
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XMI** permission to export to XML
- XMI 2.1 exported by Enterprise Architect 7.0 (or later) might not be correctly imported into earlier versions of Enterprise Architect
- When you select to apply a Data Type Definition (DTD) during an XMI 1.1 export, the UML_EA.DTD file

is written to the output directory into which the XML files are written (unless the UML_EA.DTD file is already present in the directory); no error is generated if the UML_EA.DTD file is not present in this directory during the XML export

Learn More:

- [Limitations of XML](#)^[325]
- [List of Available Permissions](#)^[205]
- [The Resources Window](#)^[789]
- [Manual Version Control with XML](#)^[334]
- [BPMN 2.0 XML](#)^[1234]

3.5.3.1.3 Import from XMI

You can import a package from an XMI (XML-based) file in any of the following formats:

- UML 1.3 (XMI 1.0)
- UML 1.3 (XMI 1.1)
- UML 1.4 (XMI 1.2)
- UML 2.0 (XMI 2.1)
- UML 2.1 (XMI 2.1)
- MOF 1.3 (XMI 1.1)
- MOF 1.4 (XMI 1.2)

You can also import the *.emx and *.uml2 files generated by tools such as Rational Software Architect (RSA) and Rational Software Modeler (RSM).

Access: Click on target package in **Project Browser**, then:

Project | Model Import/Export | Import Package From XMI File or
Right-click | Import/Export | Import Package From XMI

Use to:

- Move Enterprise Architect Model elements between models, for distributed development, manual version control and other benefits

How to:

To export a package to XMI, follow the steps below:

Step	Action	See also
1	Select the target package and Import Package From XMI menu option The Import Package From XMI dialog displays	
2	In the Filename field, type the directory path and filename from which to import the XMI file	
3	Select the Import diagrams checkbox to import diagrams	
4	Select the Strip GUIDs checkbox to remove Universal Identifier information from the file on import This enables the import of a package twice into the same model; the second import requires new GUIDs to avoid element collisions	

Step	Action	See also
5	Select the Write log file checkbox to write a log of import activity (recommended) The log file is saved in the directory from which the file is being imported, with the name <i>import.log</i>	
6	Import using single transaction defaults to selected If the import encounters locking issues, or if you are importing a large XMI file, deselect the checkbox to import the data items separately and identify problem items without blocking the whole import	
7	If you are importing from Rose XMI 1.1, click on the Treat Imported Datatypes drop-down arrow and select the datatypes to add to the model	
8	Click on the Import button Enterprise Architect indicates the progress of the import in the Import Progress field.	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Import XMI** permission to import packages from XMI
- When you import an XML file over an existing package, ALL information in the current package is deleted first; before you import the XML file, please make sure you do not have important changes that you do not want to lose
- If you are importing an XMI 1.1 file that was previously exported with a UML_EA.DTD file, the UML_EA.DTD file must be present in the directory into which the XML file is being written; an error occurs if the UML_EA.DTD file is absent

Learn More:

- [List of Available Permissions](#) ²⁰⁵
- [The Resources Window](#) ¹⁷⁸⁹
- [Manual Version Control with XMI](#) ³³⁴
- [The UML DTD](#) ³²⁶
- [Import EMX/UML2 Files](#) ³²⁴

3.5.3.1.4 Import EMX/UML2 Files

Rational Software Architect (RSA) enables you to add many UML models under a single root. These models can have cross references between them.

However, RSA cannot save the entire root as one file; it saves each UML model as a separate EMX file. This means that an EMX file with cross-references is not self-contained as it references elements in another EMX file.

In releases earlier than release 7.0, Enterprise Architect treats each EMX file as a separate model and hence does not allow for cross-references between them. From release 7.0, Enterprise Architect enables these cross-references.

You therefore have the options of:

- Importing a single EMX/UML2 file or
- Importing a group of related EMX/UML2 files together, thereby retaining the cross-references between the different files

Access: Click on target package in Project Browser, then:

Project | Model Import/Export | Import Package From XMI File: Import EMX / UML2 Files button

or

Right-click | Import/Export | Import Package From XM File: Import EMX / UML2 Files button

Use to:

- Import Enterprise Architect Model elements from EMX/UML2 files

How to:

To import single or multiple *.emx/*.uml2 files into Enterprise Architect, follow the steps below:

Step	Action	See also
1	Select the target package and Import Package From XMI menu option The Import Package From XMI dialog displays	
2	Click on the Import EMX / UML2 Files button The Import Package from XMI dialog redisplay, formatted for .EMX.UML2 file imports	
3	Click on the (...) (Browse) button next to the Directory field The Select Import EMX / UML2 File(s) dialog displays, which enables you to select multiple files	
4	Select the file or files (use (Ctrl) + click or (Shift) + click to select several files) and click on the Open button The Import Package from XMI dialog redisplay; the Selected File(s) panel lists the selected files	
5	Select the Write Log File checkbox to write a log of import activity (recommended) The log file is saved in the directory from which the file is being imported, with the name <i>import.log</i>	
6	Click on the Import button Enterprise Architect indicates the progress of the import in the Import Progress field.	

3.5.3.1.5 Limitations of XMI

Whilst XMI is a valuable means of specifying a UML model in a common format, it is relatively limited in the amount of additional information it can tolerate using the standard syntax.

Topics:

Topic	Detail	See also
Notes on Enterprise Architect	A lot of information from an Enterprise Architect model must be converted to Tagged Values, which import into other modeling	

Topic	Detail	See also
	<p>systems as additional information or are ignored completely</p> <p>Enterprise Architect can both generate and read:</p> <ul style="list-style-type: none"> • XMI 1.0 and 1.1 using UML 1.3 format • XMI 1.2 using UML 1.4 format, and • XMI 2.1 using UML 2.0 and UML 2.1 format <p>Round-tripping model elements using XMI (for example, to version control or for controlled package) is only possible using <i>XMI 1.1/ UML 1.3 - Enterprise Architect</i> format, which uses the additional Tagged Values to store the UML 2.0 information</p>	
<p>Notes for Exporting to Rose and Other Tools</p>	<p>There are discrepancies in the Unisys/Rose implementation with regard to spelling mistakes and slightly different syntax to the official XMI 1.1 specification, so problems might occur</p> <p>The way packages are arranged in different models can impact successful import into other systems; experimentation is the only work around for this problem</p> <p>Some parts of the XMI import/export process do not work as expected in products like Rational Rose; for example, Note Links are not supported, and State Operations import but do not appear in diagrams</p> <p>Rational Rose only supports import of a full project, not a single package.</p> <p>For best results, it is recommended that you keep the model elements to export to Rose simple and conforming as closely as possible to the UML 1.3 specification</p>	

3.5.3.1.6 The UML DTD

When you import or export Enterprise Architect packages to XML, the import or export process can be validated using a Data Type Definition (DTD).

Guide:

Topic	Detail	See also
<p>Notes on DTDs</p>	<p>The XML parser uses the DTD to validate the correctness of the model and to check that no syntactical errors have occurred</p> <p>It is always best to use a DTD when moving packages between Enterprise Architect models as it ensures correctness of the XML output, and prevents attempted imports of incorrect XML</p> <p>Several DTDs for XMI/UML exist; the OMG defines a standard UML1.3 DTD for use in XMI 1.1, and Enterprise Architect uses an extension of this with some additional element extensions for non-standard UML types, such as testing details</p> <p>Whenever you read an XML file, the XML parser looks in the current directory for the DTD - if specified - using the DOCTYPE element in the XML file</p> <p>If the parser cannot find the DTD, it records an error and aborts processing; you must ensure the UML_EA.DTD file is in the current XML output path (generated by default)</p>	

3.5.3.1.7 Controlled Packages

Controlled packages are a powerful means of 'externalizing' parts of an Enterprise Architect model.

Guide:

Topic	Detail	See also
Using Controlled Packages	<p>A controlled package is a package configured to save and load in XML format to a named file; the XML output is UML 1.3 compliant XML, with Enterprise Architect extensions to support diagrams and additional model elements</p> <p>Package XML is standard XML-compliant output that can be loaded into any XML viewer, or used by any XML-based tool to perform manipulations and extracts, such as document or code generators</p> <p>Controlled packages appear in the Project Browser with a small colored rectangle to the left of the package icon, as shown below for the <i>CIM</i> package:</p>  <p>Using controlled packages you can support:</p> <ul style="list-style-type: none"> • Widely distributed development by having team members submit packages in the form of XML for import into a central Enterprise Architect repository • Version control, by writing model elements in XML text files suitable for version control using standard version control software <p>Using XML this way enables you to manually connect to third-party version control software outside the Enterprise Architect environment</p> <p>Enterprise Architect internally supports the configuration of version control through SCC and CVS</p> <ul style="list-style-type: none"> • Import and export of model elements between different models <p>For example, a Class library can be re-used in many models and kept up to date in target models using controlled packages, reloading packages as required when new versions of the Class model become available</p>	<p>Version Control^[242]</p>

Notes:

- If you are importing an XML 1.1 file that was previously exported with a UML_EA.DTD (Data Type Definition) file, the UML_EA.DTD file must be present in the directory into which the XML file is being written; an error occurs if the UML_EA.DTD file is absent
- When you select to apply a DTD during an XML 1.1 export, the UML_EA.DTD file is written to the output directory into which the XML files are written (unless the UML_EA.DTD file is already present in the directory); no error is generated if the UML_EA.DTD file is not present in this directory during the XML export

3.5.3.1.7.1 Controlled Package Menu

Access: Project Browser > package | Package Control

This menu is for a package that is not itself under version control (but that might contain child packages that are under version control).

Action	Usage	Shortcut	See also
Configure (various settings for the package)	Display the Package Control Options dialog, which enables you to specify whether this package (and its child packages) is controlled and which file it is controlled through	Ctrl + Alt + P	Configure Packages [329]
Manage Baselines	Create a Baseline of the current package, or compare the current package with a previous Baseline	Ctrl + Alt + B	Managing Baselines [317]
Check In Branch	For the selected branch of the model, (that is, the selected package and all of its child packages) display the Select Packages to Check In dialog, listing all version controlled packages within that branch that are checked out to you You can then select packages in the displayed list, to be submitted for check-in.		Check in a Model Branch [283]
Check Out Branch	For the selected package, check out the package and recursively check out all of its contained sub-packages Enterprise Architect retrieves the latest version of the packages from the central repository, overwriting the current packages After check out, the packages are available for editing		Check Out a Model Branch [284]
Save package to file	Save a controlled package to an XML file	Ctrl + Alt + S	Save a Package [337]
Load package from file	Load a previously-saved XML file	Ctrl + Alt + L	Load a Package [332]
View package XML	Display the package XML after the package has been exported to XML	Ctrl + Alt + X	
Compare with XML file	(Package not under version control) Compare the current package with a previously-saved XML file of the package		The Compare Utility (Diff) [313]
Add Branch to Version Control	Apply version control to all packages within a selected model branch, in a single operation		Apply Version Control to Branches [289]
Export as Model Branch	Export a newly created model branch from your own private copy of a model		Export Controlled Model Branch [290]
Import a Model Branch	Retrieve a model branch and import it into either the source model or another model		Import Controlled Model Branch [297]
Get package (for	Enables you to gain access from packages in the		Version Control [242]

Action	Usage	Shortcut	See also
version control)	version-controlled repository that is currently available in your model		
Get All Latest (for version control)	Retrieve the latest version of the package from the repository. Available only for packages that are checked in The alternative option Get Latest - if displayed - is not intended for sharing .EAP files and should only be used when users have their own individual databases		Version Control ^[242]
Re-synch Status With VC Provider	(Version controlled package) Update the version control status value recorded for the selected package in the Enterprise Architect project to match the value reported by the version control provider, without performing an XML import or export		
Version Control Settings	Display the Version Control Settings dialog		Version Control Settings Dialog ^[270]
Update Package Status	Provide a bulk update on the status of a package. This includes status options such as Proposed , Validate and Mandatory		Update Package Status ^[411]

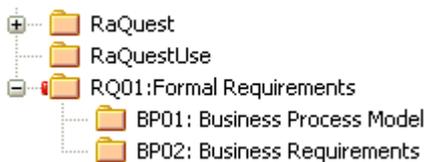
Learn More:

- For a package that is directly under version control, see [Package Version Control Menu](#)^[278]

3.5.3.1.7.2 Configure Packages

Before you can use a controlled package, you must configure it with options such as the filename to save to/load from, the type of export and the version number.

Once a package is configured and marked as controlled, it is displayed in the **Project Browser** with a small colored rectangle next to the package icon, indicating it is a controlled package; for example, the *RQ01: Formal Requirements* package below.



Access: **Project Browser | Package Context Menu | Package Control | Configure**

How to:

To configure a controlled package, follow the steps below:

Step	Action	See also
1	Select the package to control or configure, and the Configure context menu option The Package Control Options dialog displays	
2	Select the Control Package checkbox to indicate that this is a controlled package	
3	Click on the Version Control drop-down arrow and select the version control repositories; this connects the package to a specific version control configuration	
4	In the XMI Filename field, type or browse for the path and XML file for importing and exporting XML files The field accepts <i>Local Path Substitution</i> strings; for example, use an XML local path definition where: <code>myLocal Path=" C: \ Documents and Settings\ John\ Desk top\ EA Model s "</code> Then %myLocal Path% CI M. xml is equivalent to <code>C: \ Documents and Settings\ John\ Desk top\ EA Model s\ CI M. xml</code>	
5	In the UML/XMI Type field, click on the drop-down arrow and select the type of XML generated; options include: <ul style="list-style-type: none"> • Enterprise Architect XMI/UML 1.3 • Rational Rose/Unisys UML 1.3 and • Generic XMI 1.0/UML 1.3 Currently only Enterprise Architect UML 1.3 is supported for complete import/export round tripping of packages	
6	In the Version ID field, type the version ID number	
7	In the Owner field, type or select the name of the package owner	
8	If required, click on the Use DTD checkbox to use a Data Type Definition (DTD)	
9	If required, click on the Log Import/Export checkbox to log import and export activity to a log file	
10	If required, click on the Batch Import checkbox to mark the package as a Batch Import package	
11	If required, click on the Batch Export checkbox to mark the package as a Batch Export package	
12	If required, click on the Include sub-package checkbox to deselect it, to include only the immediate contents of the package in an XML export (XML stubs); this is available only for an XML 1.1/UML 1.3 export If you leave the checkbox selected, the entire sub-package hierarchy of this branch is included in the export.	
13	Click on the OK button to set the Package Control options.	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Packages** permission to configure controlled packages and package properties

- For batch import, the file date of the XML file is stored; you can bypass the batch import if there is no change; that is, the file date of the last import matches that of the current file

Learn More:

- [List of Available Permissions](#)^[205]

3.5.3.1.7.3 Remove Package from Control

If required, you can remove the control from a package. Before removing the package control, you must check in the package if it is being used for version control.

Access: [Project Browser](#) | [Package Context Menu](#) | [Package Control](#) | [Configure](#) (Ctrl+Alt+P)

How to:

To remove control from a package, follow the steps below:

Step	Action	See also
1	Select the package and the Configure menu option The Package Control Options dialog displays	
2	Deselect the Control Package checkbox If the package is under version control, this sets the Version Control field to (None)	
3	Click on the OK button to remove package control Package control for the selected package has now been removed	

Notes:

- When disconnecting a package from version control, the association between the package and the exported XML file is removed from your model; however, the XML file itself is not removed from version control, nor is it deleted from your local version control working copy folder

This is because it is possible for another model to be using the version controlled package and still referencing the associated version controlled XML file

Learn More:

- [Check In a Package](#)^[285]

3.5.3.1.7.4 Save a Package

Once you have correctly configured a package, you can save the controlled package to the designated XML file.

Access: [Project Browser](#) | [Package Context Menu](#) | [Package Control](#) | [Save Package to File](#)

How to:

To save a controlled package to file, follow the steps below:

Step	Action	See also
1	Select the package and the Save package to File menu option The export process executes automatically according to your configured preferences, overwriting any existing file	

Notes:

- If you are using a version control package in conjunction with the exported package files, you must check out the XMI file first to enable Enterprise Architect to overwrite the existing version

Learn More:

- [Configure Packages](#) ^[329]

3.5.3.1.7.5 Load a Package

If a package has been marked for control it is displayed in the Project Browser with a red rectangle to the left of the package icon. If you have previously saved the controlled package, you can reload it.

Access: [Project Browser](#) | [Package Context Menu](#) | [Package Control](#) | [Load package from file](#)

How to:

To load a controlled package from an XMI file, follow the steps below:

Step	Action	See also
1	Select the package and the Load package from file menu option If you have configured the package control details, Enterprise Architect prompts you to confirm the import.	
2	Click on the Yes button to confirm the import The current package is deleted and the saved package is imported	

Notes:

- Importing deletes the current package entirely from the model, and the action cannot be undone; you must be careful not to lose any current changes or information

Learn More:

- [Configure Packages](#) ^[329]

3.5.3.1.7.6 Batch XMI Export

You can export a group of controlled packages to XMI in one step.

Access: [Project](#) | [Model Import/Export](#) | [Batch XMI Export](#)

How to:

To export a group of controlled packages to XMI, follow the steps below:

Step	Action	See also
1	Select the Batch XMI Export menu option The Batch XMI Export dialog displays	
2	Select the checkbox against each package to include in this export run Select the Select All checkbox to select all packages in the list	
3	Select the Restore Cross References Without Prompting checkbox to automatically restore any missing cross package references without displaying a prompt	Report Deletion of Cross-Package References ^[335]
3	To save this configuration as the default, click on the Save Settings button	
4	Click on the Run Export button Enterprise Architect cycles through each checked package and exports it using the options specified in the Package Control Options dialog As long as a valid filename exists, Enterprise Architect exports the package to XMI and proceeds to the next package	Configure Packages ^[329]

Notes:

- The **Restore Cross References Without Prompting** checkbox is enabled only if the option **Report Cross Package Reference Deletions (XMI 1.1)** is set on the XML Specifications page of the Options dialog
- The option is applicable only to XMI 1.1 controlled packages

Learn More:

- [XML Specifications](#) ^[436]

3.5.3.1.7.7 Batch XMI Import

You can import a group of controlled packages from XMI into Enterprise Architect in one step.

As Enterprise Architect processes each package, it updates the **Status** column against each package name on the Batch XMI Import dialog.

- If the import is successful, Enterprise Architect updates the package status to **Imported**
- If the import is unsuccessful, Enterprise Architect updates the package status to **Not Imported**

Common reasons for an import to fail include:

- The package not being correctly configured
- The last import file date matches the import date of the file currently on disk

Access: **Project | Model Import/Export | Batch XMI Import**

How to:

To import a group of controlled packages from XMI, follow the steps below:

Step	Action	See also
1	Select the Batch XMI Import menu option The Batch XMI Import dialog displays	
2	Select the checkbox against each package to include in the import Select the Select All checkbox to select all packages in the list	
3	To save this configuration as the default, click on the Save Settings button	
4	Click on the Run Import button Enterprise Architect cycles through each checked package and imports it	Configure Packages ^[329]

Notes:

- To avoid re-importing the same module multiple times, select the **Check file date before import** checkbox; Enterprise Architect then does not import a file if the last import file date matches that of the one currently on disk

3.5.3.1.7.8 Manual Version Control with XMI

You can use XMI to support version control by writing model elements in XML text files suitable for use with standard version control software.

Using XMI in this manner enables you to manually connect to third-party version control software outside the Enterprise Architect environment.

Enterprise Architect internally supports the configuration of version control through SCC and CVS configurations.

How to:

Before using XMI for version control, follow the steps below:

Step	Action	See also
1	Select suitable packages in the Project Browser, to be marked as controlled packages	
2	Configure these with filenames that are visible to a version control system of your choice	Configure Packages ^[329]
3	Save the controlled packages to establish a model base and check these into the version control system	Save a Package ^[331]

To apply version control, follow the steps below:

Step	Action	See also
1	Continue working on a package until versioning is required	
2	Check out the package XMI file from the version control system	
3	Save the relevant package using the controlled package support	
4	Check the package back into the version control system	

To recover an earlier version, follow the steps below:

Step	Action	See also
1	Save the current version first This is important, because the package is completely deleted during the import process If necessary, manually update the version control system	
2	Get the required package version from the version control system	
3	Select the package to reload	
4	Select the Package Control Load package from file menu option to import the previous version Enterprise Architect deletes the controlled package and restores the previous version	Load a Package [332]

Learn More:

- [Version Control](#) [242]

3.5.3.1.7.9 Report Deletion of Cross Package References

When saving an XMI 1.1 controlled package, if the **Report Cross Package Reference Deletions (XMI 1.1)** option is set in the Options dialog, Enterprise Architect compares the package being exported with the latest XMI 1.1 revision of the package.

Topics:

Topic	Detail	See also
Scenario	<p><i>Master.eap</i> contains two controlled packages - <i>Package1</i> and <i>Package2</i></p>	

Topic	Detail	See also
	<p>You now import just the controlled package, <i>Package2</i>, into a new .eap file, <i>New.eap</i></p> <p>Since <i>New.eap</i> does not contain <i>Package1</i> (and hence its element <i>Class1</i>), Enterprise Architect cannot show <i>Class1</i> and the Association connector on the diagram <i>Package2</i></p> <p>When the package <i>Package2</i> is exported to XML from <i>New.eap</i>, the XML file does not contain the information that the diagram <i>Package2</i> used to contain the element <i>Class1</i> and the Association connector</p> <p>If you import this XML file into <i>Master.eap</i>, Enterprise Architect removes the element <i>Class1</i> and the Association connector from the diagram <i>Package2</i></p>	
Identify Cross-Package References	<p>In order to retain such cross-package references in XML 1.1, you can set the Report Cross Package Reference Deletions (XML 1.1) option so that when you save a controlled package, Enterprise Architect compares the latest XML 1.1 revision of the package with the package being exported</p> <p>Any items that are missing from the package are listed on the Missing Cross References dialog</p>	XML Specifications <small>[436]</small>
Actions	<p>To merge missing items, as identified in the Missing Cross References dialog, into the XML file that is being saved:</p> <ul style="list-style-type: none"> • Select the checkbox against each item type • Click on the OK button <p>If you do not select any items, they are omitted from the saved XML file</p> <p>To abort or cancel this XML Export (or Check-in, or Put Latest) operation</p> <ul style="list-style-type: none"> • Click on the Cancel button <p>To review the details of an entry:</p> <ul style="list-style-type: none"> • Double-click on the type <p>The details of each item are shown on the Missing <item type> dialog</p> <p>You can copy a value from any of the columns in this dialog to the clipboard:</p> <ul style="list-style-type: none"> • Click on the column value to highlight it; the selected value is shown in the bottom left corner of the dialog • Copy that value to the clipboard (Ctrl+C) 	

3.5.3.2 CSV Import and Export

Enterprise Architect enables you to import and export information about Enterprise Architect elements to and from CSV text files.

Topics:

Topic	Detail	See also
Specifications	To import and export element data from Enterprise Architect using CSV files, you must first set up one or more file specifications You do this because the specification defines what types of value from the spreadsheet are to be imported, and how the information is translated between the spreadsheet and Enterprise Architect	CSV Specifications ^[338]
Import From CSV	Once you have defined a CSV import specification, you can read in major element characteristics from a CSV text file	CSV Import ^[342]
Export To CSV	Once you have defined a CSV export specification it is possible to write out major element characteristics to a CSV text file	CSV Export ^[340]

3.5.3.2.1 Compare Projects

It is often useful to compare the size and row counts of two projects; for example, after:

- a database crash
- importing from XML, or
- deleting model elements

Enterprise Architect examines the number of rows in each database and produces a report indicating the total records in each and the difference in record count between the two. No examination is made of the data in the table.

Comparing projects this way is a convenient 'sanity check' after restoring a backup or doing a project data transfer. If discrepancies are found, you must investigate further manually.

Access: [Tools | Data Management | Project Compare](#)

Use to:

- Compare a .EAP file to another .EAP file
- Compare a .EAP file to a DBMS-based repository
- Compare two DBMS repositories

How to:

Step	Action	See also
1	Select the Project Compare menu option The Project Compare dialog displays	

Step	Action	See also
2	Select the required comparison: <ul style="list-style-type: none"> • .EAP to .EAP • DBMS to .EAP • .EAP to DBMS • DBMS to DBMS 	
3	In the Source Project and Target Project fields, type the name or connection string for the Source and Target projects	
4	Click on the Compare Projects button The results of the comparison display in the panel at the bottom of the dialog	
5	If required, click on the Print List button to print the results	

3.5.3.2.2 CSV Specifications

To import and export element data to and from Enterprise Architect using CSV files, you must first set up one or more file specifications.

A file specification defines the:

- Fields from the spreadsheet in the order they are imported or exported
- Filename (optional) and
- Delimiter between columns

Once you have defined one or more specifications, one can be selected in the CSV Import/Export dialog as the current specification to apply during an import or export action.

CSV Import/Exports only imports and exports *elements* (within packages) and their properties; items such as Class attributes cannot be imported or exported through this mechanism.

XML import/export provides a solution to this limitation, as does use of the Automation Interface (Object Model).

Access: [Project | Model Import/Export | CSV Import/Export Specifications](#)

How to:

To define a specification:

- Select the **CSV Import/Export Specifications** menu option
- The CSV Import/Export Specification dialog displays; complete the fields as required, as indicated in the following table:

Field	Usage	See also
Specification Name	Select the unique name for this specification	
Delimiter	Specify the character delimiter to use between record fields If a field contains an instance of the delimiter, the field is exported wrapped in " (quotation marks) and all instances	

Field	Usage	See also
	of " in the field are doubled (that is, " becomes """)	
Notes	Record a brief description of the specification	
Default Filename	Select the default filename	
Default Direction	Set the default action - Import or Export A specification can be used in either direction, but this enables you to set the default type	
Default Types	Limit the element types being exported, by entering a comma-separated list: for example: <i>class,requirement,component,node,object</i> . <ul style="list-style-type: none"> • If you specify element types, ONLY elements of those types are exported or imported; therefore, in order to enable the Preserve Hierarchy option to operate (if selected) you must include Package as an element type, otherwise there are no packages in which to preserve the hierarchy • If you do not specify any default element types, all elements including Packages are exported or imported and the hierarchy can be preserved 	
Preserve Hierarchy	Include fields generated by Enterprise Architect to embed/reconstruct the package hierarchy	Using Preserve Hierarchy ^[340]
Available Fields	Select from a list of possible record fields, not yet allocated	
File Specification	List the record fields (in the order they are plotted across the spreadsheet) already assigned	
Add Field	Move all selected fields in the top list to the bottom list	
Remove Field	Move all selected fields in the bottom list back to the available list	
New	Create a new specification	
Save	Save changes to the currently selected specification	
Save As	Save the current specification with a new name	
Delete	Delete the current specification	
Close	Close this dialog	

Notes:

- In **Available Fields** and **File Specification**, the record fields **Created Date** and **Modified Date** are not set when imported from CSV

Learn More:

- [XML Import and Export](#)^[319]
- [Enterprise Architect Object Model](#)^[1822]

3.5.3.2.2.1 Using Preserve Hierarchy

The **Preserve Hierarchy** option on the CSV Import/Export Specification dialog includes fields generated by Enterprise Architect to embed/reconstruct the package hierarchy in a CSV import or export.

When selected, the **Preserve Hierarchy** option inserts two fields into the CSV specification that are:

- automatically populated by Enterprise Architect on export and
- used to reconstruct the exported package's hierarchy upon import.

Field	Usage	See also
CSV_KEY	A unique identifier for the current element This key is unique per export; subsequent exports produce different keys for the same set of elements	
CSV_PARENT_KEY	The corresponding CSV_KEY of the current element's parent If the field is left blank or references a non-existent CSV_KEY, the element is added to the top level of the package	

It is highly recommended that you do not change these fields if they have been automatically generated by Enterprise Architect's CSV exporter.

If you intend to import hierarchical information from a spreadsheet that was **not** populated by exporting data from Enterprise Architect, you must add these two fields to your spreadsheet as the last two columns, and populate the columns yourself.

For example:

NAME	TYPE	NOTES	PRIORITY	STATUS	CSV_KEY	CSV_PARENT_KEY
Requirement						
Package	Package	Notes Package1			Package1	
REQ1	Requirement	Notes on REQ1	High	Approved	REQ1	Package1
REQ2	Requirement	Notes on REQ2	High	Approved	REQ2	Package1
REQ2.1	Requirement	Notes on REQ2.1	High	Approved	REQ2.1	REQ2
REQ2.2	Requirement	Notes on REQ2.2	Med	Approved	REQ2.2	REQ2
REQ2.3	Requirement	Notes on REQ2.3	High	Approved	REQ2.3	REQ2
REQ3	Requirement	Notes on REQ3	High	Approved	REQ3	Package1
REQ3.1	Requirement	Notes on REQ3.1	High	Approved	REQ3.1	REQ3
REQ3.2	Requirement	Notes on REQ3.2	High	Approved	REQ3.2	REQ3
REQ4	Requirement	Notes on REQ4	High	Approved	REQ4	Package1
REQ4.1	Requirement	Notes on REQ4.1	High	Approved	REQ4.1	REQ4
REQ4.2	Requirement	Notes on REQ4.2	High	Approved	REQ4.2	REQ4
REQ4.3	Requirement	Notes on REQ4.3	High	Approved	REQ4.3	REQ4

3.5.3.2.3 CSV Export

Enterprise Architect enables you to export information about elements to a CSV text file, using a CSV export specification.

If you intend to *re-import* the exported information into Enterprise Architect at some point, it is recommended that you include the **GUID** field in the CSV export specification. This ensures that Enterprise Architect can identify and update existing elements, rather than creating duplicates.

Access: [Project Browser](#) | [Package Context Menu](#) | [Import/Export](#) | [CSV Import/Export](#)

How to:

To export data in CSV format, follow the steps below

1. In the Project Browser, right-click on the package containing the elements to export and select the **CSV Import/Export** menu option
2. The CSV Import/Export dialog displays; complete the fields as required, as indicated in the following table:

Field	Usage	See also
Package	Confirm the name of the current selected package	
Specification	Specify the name of the export specification ^[320] to use	CSV Specifications ^[338]
Edit/New	Edit the export specification or create a new one	
File	Specify the filename to export to	
Types	<p>List the element types to export: leave blank for all, or enter a comma-separated list of types</p> <p>If you specify element types, ONLY elements of those types are exported; therefore, to enable the Preserve Hierarchy option in the specification to operate (if selected) you must include <i>Package</i> as an element type, otherwise no Packages are exported in which to preserve the hierarchy</p> <p>If you do not specify any element types, all elements including Packages are exported and the hierarchy can be preserved</p>	
Action	Select the Export radio button to export to file	
Print Results	Print out the result list	
View File	View the resultant CSV file with the default Windows application for CSV files	
Run	Perform the export	
Close	Exit this dialog	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have both **Export XMI** and **Import XMI** permissions to use the **CSV Import/Export** option

Learn More:

- [List of Available Permissions](#) ^[205]

3.5.3.2.4 CSV Import

Enterprise Architect enables you to import information about elements from a CSV text file, using a CSV export specification.

You import the CSV file into a selected package; if this package or any element within the package has a lock on it, you cannot import the CSV file into it. The **Import** option on the dialog is grayed out.

The format and content of the source data file should resemble the following:

The screenshot shows a spreadsheet window titled 'req.xls - OpenOffice.org Calc'. The spreadsheet contains a table with 9 rows and 9 columns. The columns are labeled A through H. The rows contain requirement data.

	A	B	C	D	E	F	G	H
1	Name	Type	Notes	Phase	Version	Status	Priority	Difficulty
2	FR27 – Retrieval of histo	Requirement	The system must be able to	1	1	Approved	Medium	Medium
3	FR28 – Transmission of	Requirement		1	1	Approved	Medium	Medium
4	FR29 – The staff must no	Requirement	The system must use stan	1	1	Mandatory	Low	Medium
5	FR30 – The Interface mu	Requirement	Critical to the success of th	1	1	Approved	High	Medium
6	FR44 – The interface sho	Requirement	An important aspect of the	1	1	Approved	High	Medium
7	FR5 – 2000 hours mean	Requirement	The mean time between fai	1	1	Proposed	Medium	Medium
8	FR6 – Must be recovera	Requirement	In the event of software or h	1	1	Approved	Medium	High
9	FR7 – 99.999% accurac	Requirement	The system accuracy defin	1	1	Approved	Medium	Medium

Access: Project Browser | Package Context Menu | Import/Export | CSV Import/Export

How to:

To import data in CSV format, follow the steps below:

1. In the Project Browser, right-click on the package to import into and select the **CSV Import/Export** context menu option
2. The CSV Import/Export dialog displays; set the required options; as outlined below:

Field	Usage	See also
Package	Confirm the name of the current selected package	
Specification	Specify the name of the import specification <input type="text" value="1705"/> to use	Import specification <input type="text" value="338"/>
Edit/New	Edit the import specification or create a new one	
File	Specify the spreadsheet filename to import from	
Types	Not used for import	
Action	Select the Import radio button to import from the file (grayed-out if the package or a child item in the package is locked)	
Print Results	Print out the result list	
View File	View the source CSV file with the default Windows application for CSV files	
Run	Perform the import	

Field	Usage	See also
Close	Exit this dialog	

Notes:

- When importing, Enterprise Architect checks the specification to see if there is a GUID field included; if there is, Enterprise Architect attempts to locate the element identified by the GUID and, if successful, updates the current element rather than creating a new one
- If no GUID field is defined, or Enterprise Architect cannot locate the identified element, a new element is created and placed in the current package; during import, **Type** is a mandatory field in the source file and must match one or more of the legal Enterprise Architect element types: for example: Requirement, or Class
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have both **Export XMI** and **Import XMI** permissions to use the **CSV Import/Export** option

Learn More:

- [List of Available Permissions](#) ^[205]
- [Enterprise Architect element types](#) ^[188]

3.5.3.3 Perform a Project Data Transfer

The Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect support a range of data repositories.

At some point, it might become necessary to move a complete model from one repository to another, row by row, table by table.

Access: **Tools | Data Management | Project Transfer**

Use to:

- Upload an existing .EAP file to a SQL Server or MySQL repository
- Download a repository in MySQL or SQL Server to a .EAP file
- Move a repository from SQL Server to MySQL or from one server to another
- Move all records from a .EAP file with replication to a model with none (Remove Replication)
- Copy all records from a .EAP file to another (recommended after serious network crash or repeated database corruption)
- Copy all records from a JET 3.5 to JET 4 (Access 2000 or XP) repository - or back the other way

How to:

To perform a project data transfer, follow the steps below:

Step	Action	See also
1	Take a backup of the target project to ensure that you can recover any important information it contains	
2	Select the Project Transfer menu option The Project Transfer dialog displays	

Step	Action	See also
3	Click on the option for the required transfer type: <ul style="list-style-type: none"> • .EAP to .EAP • DBMS to .EAP • .EAP to DBMS • DBMS to DBMS 	
4	In the Source Project and Target Project fields, type or select the name or connection string for the Source and Target projects	
5	If you want to capture the transfer in a log file, select the Logfile checkbox and browse for the appropriate log file location	
6	Click on the Transfer button	
7	It is good practice to do a Project Compare after this process to verify that all records are written	Compare Projects ^[337]

Notes:

- All records in the target repository are overwritten
- You cannot move a model from a source .EAP file of a version earlier than 3.5.0
- You must transfer data into a repository (where scripts have been run to set up tables), not just a database; if necessary, follow the steps below:
 - Install the DBMS software and create a database; ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic
 - Run a script supplied by Sparx Systems to create the required tables
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Transfer Data** permission to transfer project data between repositories

Learn More:

- [Sparx Systems Scripts webpage](#)
- [Create a Repository](#) ^[148]
- [List of Available Permissions](#) ^[205]

3.5.3.3.1 Copy Packages Between Projects

Using the XML import/export capabilities of Enterprise Architect, you can copy and move packages between Enterprise Architect projects.

This gives you a high level of flexibility in building a project from re-usable parts and from elements produced in widely-dispersed geographic regions.

Access: [Project Browser](#) | [Package Context Menu](#) | [Import/Export](#) | [Export package to XMI file](#)
[Project Browser](#) | [Package Context Menu](#) | [Import/Export](#) | [Import package from XMI file](#)

Use to:

- Copy individual child packages within or between models
- Duplicate larger structures, such as complete models or projects

How to:

To copy a package from one Enterprise Architect project to another, follow the steps below:

Step	Action	See also
1	Open the Enterprise Architect project to copy <i>from</i>	
2	In the Project Browser, right-click on the package to copy and select the Export package to XMI file option The Export Package to XMI dialog displays	
3	Select the appropriate options and filename	Export to XMI ^[320]
4	Click on the Export button to begin the export process	
5	When the export is complete, open the Enterprise Architect project to copy <i>into</i>	
6	In the Project Browser, navigate to the location to import the package into	
7	Right-click and select the Import package from XMI file option The Import Package from XMI dialog displays	
8	Select the appropriate options and filename	Import from XMI ^[323]
9	Click on the Import button The package is copied from the source project to the destination project.	

Notes:

- If the package you are importing already exists in the target project (that is, it has been imported previously), you must either import *over* the existing package or select the **Strip GUIDs** option

In the latter case, Enterprise Architect creates a copy of the original package

Learn More:

- [Copy a Package](#) ^[535]
- [Copy Elements Between Packages](#) ^[634]

3.6 Project Management



Enterprise Architect provides strong support for:

- Project Management, in estimating project size, measuring risk and effort, and assigning resources to elements
- Change control and maintenance

Topics:

Topic	Detail	See also
Metrics and Estimation	<p>Project estimation is working out how much time and effort is required to build and deploy a solution</p> <p>Enterprise Architect provides the Use Case metrics facility as a means of:</p> <ul style="list-style-type: none"> • Measuring the complexity of a system • Getting an indication of the effort required to implement the model • Getting an indication of the project timescale <p>You base these estimates on carefully-calibrated metrics</p>	Use Case Estimation ^[400]
Resource Management	<p>Resources are the people who work on a project</p> <p>You can assign roles to resources and allocate tasks on specific model elements, which enables tracking of effort and estimation of time to complete</p>	Project Resources ^[348]
Project Maintenance	<p>During a project you monitor and manage the development and progress of individual model elements</p> <p>You can record problems, changes, issues and tasks that affect these individual elements as they arise, and document the solution and associated details</p> <p>Similarly, Enterprise Architect helps you to manage changes and issues that apply to the whole system</p>	Maintenance ^[1713] Changes and Defects ^[1719]
Project Tasks and Issues	<p>In the course of a project, there are various non-technical tasks that are vital to the successful management and completion of the project, such as meetings</p> <p>Enterprise Architect helps you to record and monitor these, and to manage non-technical project issues as they arise</p>	Project Tasks ^[356] Project Issues ^[358] Project Task Allocation ^[366]

Learn More:

- [The Project Management Window](#) ^[347]
- [Personal Information](#) ^[377]
- [The Project Information Window](#) ^[356]

- [Project Calendar](#)^[390]
- [Project Glossary](#)^[362]
- [Update Package Status](#)^[411]
- [Manage Bookmarks](#)^[472]
- [Monitor Change Events](#)^[413]

3.6.1 The Project Management Window

The Project Management window enables you to input the following quantities associated with an element contained in the model:

- Resources - the people who work on a project, who can be assigned roles and allocated tasks
- Effort - effort expended in work on the element
- Risks - risk associated with the element
- Metrics - metrics measured for an element

Each of these four quantities has a separate tab in the window.

Access: [View](#) | [More Element Tools](#) | [Project Management](#)

Reference:



Action	Usage	See also
New	Create new item	Resource Allocation ^[349] Effort Management ^[350] Risk Management ^[351] Metrics ^[352]
Save	Save changes to an item	
Save As New	Duplicate an existing entry You must change an item's Role for this to become enabled	
Delete	Delete an item from the list	
Sort	Sort the items in the list into alphabetical order	
Print	Print item data from the list	
Browse Element	Display the Element Browser window for the selected element, to list and select the project management items for the element	The Element Browser ^[687]
Show/Hide Properties	Swap between detailed and summary window styles	
Help	Show help contents for this window.	

Notes:

- Click on an element in the Project Browser to display project management information for that element in the Project Management window
- The window has two formats:
 - *List* mode - provides a detailed list showing several columns of information
 - *Item* mode - a summary list plus an editor to add or edit information for a single selected item
- Toggle between these modes using the **Show/Hide Properties** button in the window toolbar; each tab toggles between Item mode and List mode independently
- An asterisk on a tab name indicates that the tab contains saved information; if there is no information for a category of item, or the information has not yet been saved, its tab has no asterisk
- Right-click on the list to view the context menu, which enables you to add and delete list items

3.6.2 Project Resources

Resources are the people who work on a project. They can be assigned roles and allocated tasks, which enables tracking of effort and estimation of time to complete.

Resources are added, modified and deleted using the Project Management window.

Access: [View](#) | [More Element Tools](#) | [Project Management](#) (Ctrl+Shift+7)

Use to:

Usage	See also
Allocate a resource to an element	Resource Allocation ^[349]
Record additional project management information for an element	Effort Management ^[350] Risk Management ^[351] Metrics ^[352]
Obtain a report of resource allocation details	Resource Report ^[370]
Configure Project Management data and populate the drop-down lists used on the Project Management dialog tabs	Roles ^[780] Clients ^[782] Effort Types ^[353] Metric Types ^[354] Risk Types ^[355]

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Project Information** permission to update and manage project resources, effort, metrics and risks

Learn More:

- [The Project Management Window](#) ^[347]
- [List of Available Permissions](#) ^[205]

3.6.2.1 Resource Allocation

In Enterprise Architect, the Project Manager can connect a named resource in a named role to a given model element.

To select the element to which to allocate resources, click on the required element in the Project Browser.

Access: **View | More Element Tools | Project Management > Resource Allocation: New (Ctrl+Shift+7)**

Use to:

- Track how far development of required components and Classes has progressed

How to:

To enter resource allocation details for an element, follow the steps below:

Step	Action	See also
1	Select the element in the Project Browser	
2	Select the Project Management menu option The Project Management window displays, showing the Resource Allocation tab	The Project Management Window ³⁴⁷
3	Click on the New icon on the Project Management window toolbar	
4	On the Resource Allocation tab enter the: <ul style="list-style-type: none"> • Name of the resource (click on the drop-down arrow and select, or type the name in) • Role of the resource (click on the drop-down arrow and select, or type the name in) • Start and end date for the availability of the resource • Time allocated to the resource • Percentage of the task the resource has completed • Expected time allocated to the resource • Actual time expended by the resource • Description of the work being done by the resource (this text is also displayed in the Notes window; it cannot be edited in that window) • Notes on the activity history of the work done by the resource (this text is also displayed in the Notes window; it cannot be edited in that window) 	Notes ⁷⁶⁹
5	Click on the Save icon in the Project Management window toolbar	

Notes:

- To edit existing Resource Allocation items for this element, click on the required item in the:
 - List panel to the left of the window, in *Item* mode
 - List, in *List* mode
 - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; resource allocation item icons have an **R** in the bottom right corner

Learn More:

- [The Element Browser](#)^[687]

3.6.2.2 Effort Management

In Enterprise Architect, the Project Manager can allocate effort (as time) to work on a given model element.

To select the element to which to allocate effort, click on the required element in the Project Browser.

Access: **View | More Element Tools | Project Management > Effort: New (Ctrl+Shift+7)**

Use to:

- Enter the effort allocated to an element

How to:

To enter effort allocation details for an element, follow the steps below:

Step	Action	See also
1	Select the element in the Project Browser	
2	Select the Project Management menu option The Project Management window displays, showing the Resource Allocation tab	
3	Click on the Effort tab	
4	Click on the New icon on the Project Management window toolbar	
5	On the Effort tab enter the: <ul style="list-style-type: none"> • Name of the effort (a short description) • Type of the effort (type the name or click on the drop-down arrow and select; the selection list is drawn from the global Effort Type list, but any new efforts you type in the Type field are not added to the list) • Time the effort is expected to expend • Notes on effort (this text is also displayed in the Notes window; it cannot be edited in that window) 	Effort Types ^[353] Notes ^[769]
6	Click on the Save icon in the Project Management window toolbar	

Notes:

- To edit existing Effort items for this element, click on the required item in the:
 - List panel to the left of the window, in *Item* mode
 - List, in List mode, or
 - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; effort item icons have an **E** in the bottom right corner
- Although Enterprise Architect does not currently provide detailed reports on effort within a model, you can use the Automation Interface or similar tools to create your own custom reports based on effort information you enter

Learn More:

- [The Element Browser](#)^[687]

- [Enterprise Architect Object Model](#)^[1822]

3.6.2.3 Risk Management

In Enterprise Architect, the Project Manager can allocate the possible weighting of defined risks to work on a given model element.

To select the element to which to allocate risk weightings, click on the required element in the Project Browser.

Access: [View](#) | [More Element Tools](#) | [Project Management](#) > [Risks: New](#) (Ctrl+Shift+7)

Use to:

- Enter the details of risks that might impact work on an element

How to:

To enter risk details for an element, follow the steps below:

Step	Action	See also
1	Select the element in the Project Browser	
2	Select the Project Management menu option The Project Management window displays, showing the Resource Allocation tab	
3	Click on the Risks tab	
4	Click on the New icon on the Project Management window toolbar	
5	On the Risks tab enter the: <ul style="list-style-type: none"> • Name of the risk (a short description) • Type of the risk (Mandatory: type the name or click on the drop-down arrow and select; the selection list is drawn from the global Risk Type list, but any new risks you type in the Type field are not added to the list) • Weighting allocated to the risk • Notes on the risk (this text is also displayed in the Notes window; it cannot be edited in that window) 	Risk Types ^[355] Notes ^[769]
6	Click on the Save icon in the Project Management window toolbar	

Notes:

- The risks described here are not the same as those represented by Risk elements; the risks described above are properties of a single element, whilst the Risk element represents something that can impact a range of other elements
- If you select a global risk type from the **Type** drop-down list and the associated **Weight** field is empty, the default Weight value is allocated to the **Weight** field on the Risks tab
- To edit existing Risk items for this element, click on the required item in the:
 - List panel to the left of the window, in *Item* mode
 - List, in List mode
 - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; risk item icons have an **Ri** in the bottom right corner

- Although Enterprise Architect does not currently provide detailed reports on risks within a model, you can use the Automation Interface or similar tools to create your own custom reports based on risk information you enter

Learn More:

- [The Element Browser](#)^[687]
- [Enterprise Architect Object Model](#)^[1822]
- [Risk Elements](#)^[1298]

3.6.2.4 Metrics

In Enterprise Architect, the Project Manager can allocate the possible weighting of defined metrics to work on a given model element.

To select the element to which to allocate metric weightings, click on the required element in the Project Browser.

Access: [View](#) | [More Element Tools](#) | [Project Management](#) > [Metrics: New](#) (Ctrl+Shift+7)

Use to:

- Enter the details of metrics that you might apply to work on an element

How to:

To enter metric details for an element, follow the steps below:

Step	Action	See also
1	Select the element in the Project Browser	
2	Select the Project Management menu option The Project Management window displays, showing the Resource Allocation tab	
3	Click on the Metrics tab	
4	Click on the New icon on the Project Management window toolbar	
5	On the Metrics tab enter the: <ul style="list-style-type: none"> • Name of the metric (a short description) • Type of the metric (Mandatory: type the name or click on the drop-down arrow and select; the selection list is drawn from the global Metric Type list, but any new metrics you type in the Type field are not added to the list) • Weighting allocated to the metric • Notes on the metric (this text is also displayed in the Notes window; it cannot be edited in that window) 	Metric Types ^[354] Notes ^[769]
6	Click on the Save icon in the Project Management window toolbar	

Notes:

- If you select a global metric type from the **Type** drop-down list and the associated **Weight** field is empty, the default Weight value is allocated to the **Weight** field on the Metrics tab
- To edit existing Metric items for this element, click on the required item in the:

- List panel to the left of the window, in *Item* mode
- List, in List mode
- Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; metric item icons have an **M** in the bottom right corner
- Although Enterprise Architect does not currently provide detailed reports on metrics within a model, you can use the Automation Interface or similar tools to create your own custom reports based on metric information you enter

Learn More:

- [The Element Browser](#)^[687]
- [Enterprise Architect Object Model](#)^[1822]

3.6.2.5 Effort Types

Enterprise Architect enables you to add an effort *type* to the global list of effort types that can be added to any element in the model.

The global list of effort types displays in the **Type** field drop-down list on the Effort tab of the Project Management window.

Access: [Settings](#) | [Project Types](#) | [Project Indicators](#) > [Effort: New](#)

Use to:

- Specify the effort types used when assigning effort to an element

How to:

To add a new effort type to the global list, follow the steps below:

Step	Action	See also
1	Select the Project Indicators menu option The Project Indicators dialog displays	
2	Click on the Effort tab.	
3	Click on the New button (To edit an existing effort type, click on the effort type name in the Defined Effort Types list)	
4	Complete the fields as follows: <ul style="list-style-type: none"> • In the Effort field type the name of the effort type • In the Description field type a short description of the effort type • In the Weight field type the default weighting to apply to the effort type • In the Note field, type any additional information on the effort type 	
5	Click on the Save button.	

Notes:

- Although Enterprise Architect does not currently provide detailed reports on effort within a model, you can use the Automation Interface or similar tools to create your own custom reports based on effort information you enter
- You can transport effort types between models, using Export Reference Data and Import Reference Data

Learn More:

- [The Project Management Window](#)^[347]
- [Enterprise Architect Object Model](#)^[1822]
- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

3.6.2.6 Metric Types

Enterprise Architect enables you to add a metric *type* to the global list of metric types that can be added to any element in the model.

The global list of metric types displays in the **Type** field drop-down list on the Metrics tab of the Project Management window.

Access: [Settings](#) | [Project Types](#) | [Project Indicators](#) > [Metric: New](#)

Use to:

- Specify the metric types used when assigning metrics to an element

How to:

To add a new metric type to the global list, follow the steps below:

Step	Action	See also
1	Select the Project Indicators menu option The Project Indicators dialog displays	
2	Click on the Metric tab.	
3	Click on the New button (To edit an existing metric type, click on the metric type name in the Defined Metrics list)	
4	Complete the fields as follows: <ul style="list-style-type: none"> • In the Metric Type field type the name of the metric type • In the Description field type a short description of the metric type • In the Weight field type the default weighting to apply to the metric type • In the Note field, type any additional information on the metric type 	
5	Click on the Save button.	

Notes:

- Although Enterprise Architect does not currently provide detailed reports on metrics within a model, you

can use the Automation Interface or similar tools to create your own custom reports based on metric information you enter

- You can transport metric types between models, using Export Reference Data and Import Reference Data

Learn More:

- [The Project Management Window](#)^[347]
- [Enterprise Architect Object Model](#)^[1822]
- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

3.6.2.7 Risk Types

Enterprise Architect enables you to add a risk *type* to the global list of risk types that can be added to any element in the model.

The global list of risk types displays in the **Type** field drop-down list on the Risks tab of the Project Management window.

Access: [Settings | Project Types | Project Indicators > Risk: New](#)

Use to:

- Specify the risk types used when defining risks to an element

How to:

To add a new risk type to the global list, follow the steps below:

Step	Action	See also
1	Select the Project Indicators menu option The Project Indicators dialog displays	
2	Click on the Risk tab.	
3	Click on the New button (To edit an existing risk type, click on the risk type name in the Defined Risks list)	
4	Complete the fields as follows: <ul style="list-style-type: none"> • In the Risk Type field type the name of the risk type • In the Description field type a short description of the risk type • In the Weight field type the default weighting to apply to the risk type • In the Note field, type any additional information on the risk type 	
5	Click on the Save button	

Notes:

- Although Enterprise Architect does not currently provide detailed reports on risks within a model, you can use the Automation Interface or similar tools to create your own custom reports based on risk

information you enter

- You can transport risk types between models, using Export Reference Data and Import Reference Data

Learn More:

- [The Project Management Window](#)^[347]
- [Enterprise Architect Object Model](#)^[1822]
- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

3.6.3 The Project Information Window

The Project Information window documents tasks and issues that relate directly to the current project, and enables you to view and develop the project glossary of terms.

Access: [View](#) | [More Project Tools](#) | [Project Information](#) (Alt+2)

Use to:

Usage	See also
Review major project tasks that require attention; you can filter tasks based on their current status	Project Tasks ^[356]
Review events, occurrences and situations that impact on project development and delivery	Project Issues ^[358]
Review the technical and business terms already defined for a model You can add to the list, delete or change items, and filter the list to exclude by type	Project Glossary ^[362]

You can add or work on an item in the Project Information window by *right-clicking* (context menu) or *double-clicking* on the blank or completed item line.

Notes:

- Right-clicking in the Project Information window displays a context menu that has options for filtering tasks/issues by status, and glossary by term; you can also rearrange the sort-order by clicking in the title bar of the column that the items are to be indexed on

3.6.3.1 Project Tasks

The *Project Tasks List* is a convenient 'To Do' list of major project work items that are not recorded elsewhere, and can be used to track events such as requests or corrections. It is a tab of the Project Information window.

Notes:

- Right-click on the list to view the context menu, and select to add, modify or delete tasks, or to set a status filter
- To set the sort order, click the title-bar of the column on which to index the tasks
- To print out the currently displayed items, select the **Print List** context menu option
- You can transport task definitions between models using Export Reference Data and Import Reference Data

Learn More:

- [Add, Modify and Delete Tasks](#) ^[357]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

3.6.3.1.1 Add, Modify and Delete Tasks

This topic explains how to maintain project tasks.

Access: **View | More Project Tools | Project Information > Project Tasks**
View | Project Calendar > Project Tasks mode
View | Personal Information > Project Tasks

Use to:

- Add project tasks
- Edit project tasks
- Delete project tasks

How to:

To maintain tasks, follow the steps below:

Step	Action	See also
1	If creating a new task: <ul style="list-style-type: none"> • Double-click in a blank area of the Project Tasks tab, or on a cell of the Project Calendar If editing an existing item: <ul style="list-style-type: none"> • Double click on the item on the Project Tasks tab or Project Calendar The Task Detail dialog displays	The Project Information Window ^[356] Project Calendar ^[390]
2	Enter or update the following details of the task, selecting a value from the field's drop-down list where appropriate: <ul style="list-style-type: none"> • The task name • Auto counters - if you have configured these, click on the Auto button • The task type • The task owner • The expected start and end date for the task (select the check boxes to activate the dates) • The current status of the task • The person this task has been assigned to • The task priority: high, medium or low • The expected total time for the task and the actual time expended • The percentage complete • The phase associated with this task • A description of the task • Any progress history appropriate to the task 	Use Auto Naming and Auto Counters ^[628]
3	Click on the Apply button	

Step	Action	See also
4	To create another entry, click on the New button, or to close, click on the OK button	
5	To delete a task: <ul style="list-style-type: none"> • Right-click on the task and select the Delete context menu option • Click on the Yes button on the confirmation prompt 	

3.6.3.2 Project Issues

Any identified issues can be recorded against the current project, on either the:

- Project Issues tab of the Project Information window, or
- Project Issues dialog

For each Issue, you record the description, date, owner and status.

Access: **View | More Project Tools | Project Information > Project Issues tab (Alt+2)**
Project | Documentation | Issues

Usage	See also
Add, edit and delete Issues	Add, Delete and Modify Issues ^[358]
Generate and view a rich text format report of your issue list	Report From Project Issues Dialog ^[360] Report From Project Issues Tab ^[360] Report Output Sample ^[361]

Notes:

- You can transport these issue definitions between models, using Export Reference Data and Import Reference Data
- To print out the currently displayed items, select the **Print List** context menu option
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Issues** permission to update and delete Issues records

Learn More:

- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]
- [List of Available Permissions](#) ^[205]

3.6.3.2.1 Add, Delete and Modify Issues

This topic explains how to maintain project issue records using either the:

- Issue Detail dialog from the Project Issues tab of the Project Information window, or
- Project Issues dialog

These two dialogs have very similar fields.

Access: [View](#) | [More Project Tools](#) | [Project Information](#) > [Project Issues tab](#) (Alt+2)
[Project](#) | [Documentation](#) | [Issues](#)

Use to:

- Add project issues
- Edit project issues
- Delete project issues

How to:

To maintain issues, follow the steps below:

Step	Action	See also
1	If creating a new issue: <ul style="list-style-type: none"> • On the Project Issues tab, double-click in a blank area to display the Issue Detail dialog • On the Project Issues dialog, click on the New button If editing an existing item: <ul style="list-style-type: none"> • On the Project Issues tab, double click on the item • On the Project Issues dialog, click on the item in the Project Issues & Discussion list 	The Project Information Window ^[356]
2	Enter or update the following issue details, selecting a value from the field's drop-down list where appropriate: <ul style="list-style-type: none"> • The issue name • Auto counters - if you have configured these, click on the Auto button • The issue priority • The date the issue was raised • The issue status • The issue owner • A description of the issue • The name of the person who resolved the issue • The date on which the issue was resolved (select the check box to activate the date) • Any comments on the resolution 	Use Auto Naming and Auto Counters ^[628]
3	Click on the Apply or Save button	
4	If the issue is closed (and all the Resolution fields are completed), click on the Close Issue button	
5	To create another entry, click on the New button, or to close, click on the OK or Close (Project Issues dialog) button.	
6	To delete an issue: <ul style="list-style-type: none"> • On the Project Issues tab, right-click on the entry and select the Delete context menu option, then click on the Yes button on the confirmation prompt • On the Project Issues dialog, click on the item in the Project Issues & Discussion list, ensure that the Resolution fields are complete, then click on the Delete button 	

Notes:

- You cannot delete a Closed issue through the Project Issues dialog
- You can filter the list of issues by status, to include or exclude:
 - On the Project Issues dialog, closed issues, using the **Show Closed Issues** checkbox
 - On the Project Issues tab, all issues or just Open, Closed or Under Review issues, using the **Set Term Filter** context menu option

3.6.3.2.2 Report From Project Issues Dialog

This topic explains how to generate an RTF report on your project issue records using the Project Issues dialog.

Access: [Project | Documentation | Issues](#)

Use to:

- Generate and view an RTF report on project issues

How to:

To generate your project issues report, follow the steps below:

Step	Action	See also
1	On the Project Issues dialog, click on the Report button The Save As dialog displays	
2	Browse for and select the appropriate file location In the File name field, type the file name for the report	
3	Click on the Save button A status message displays when the report has been generated	
4	Click on the OK button and then on the View RTF button The report displays in your default viewer	Report Output Sample [361]

3.6.3.2.3 Report From Project Issues Tab

This topic explains how to generate an RTF report on your project issue records using the Project Issues tab of the Project Information window.

Access: [View | More Project Tools | Project Information > Project Issues tab \(Alt+2 \)](#)

Use to:

- Generate and view an RTF report on project issues

How to:

To generate your project issues report, follow the steps below:

Step	Action	See also
1	Right-click on the Project Issues tab; select the Create RTF Report context menu option The Save As dialog displays	
2	Browse for and select the appropriate file location In the File name field, type the file name for the report	
3	Click on the Save button A status message displays when the report has been generated	
4	Click on the OK button and then on the View RTF button The report displays in your default viewer	Report Output Sample 361

3.6.3.2.4 Report Output Sample

The following illustration is an example of the output from an *Issues* report.

List of Project Issues: 24-Jul-2010 9:47:00 AM

Issue	Date/Owner	Description	Resolution
Test servers will be delayed	24/07/2010 Eloise Norman	The test server builds have been delayed because the particular (unusual) memory requirements to match the customer's site are not available on shore. They are being sourced from Singapore but it will delay the builds and delivery of the machines.	Closed: 24/07/2010 Geoffrey Sparks The machines will be built and delivered using standard memory and the proprietary memory will be added later. All performance tests will be delayed until the memory is available.
Public Holidays	24/07/2010 Joanna Stoa	The schedule includes staff working on public holidays. A number of staff have indicated that contrary to what they stated earlier they are not available.	Open: 24/07/2010
Compiler Version disparity	24/07/2010 Eloise Norman	A number of the developers have downloaded different versions of a number of the compilers. This has lead to unpredictable builds impacting on testing.	Under Review: 24/07/2010

3.6.3.3 Project Glossary

The Project Glossary enables you to set up a list of defined terms for your project.

Access: [View | More Project Tools | Project Information > Project Glossary](#) (Alt+2)
[Project | Documentation | Glossary](#)

Usage	See also
Add, delete and modify the project glossary entries through either the: <ul style="list-style-type: none"> Project Glossary tab on the Project Information window, or Glossary dialog 	Project Glossary Tab ^[362] The Glossary Dialog ^[363]
Separate glossary items by category; for example, Business terms and Technical terms	Project Glossary Tab ^[362] The Glossary Dialog ^[363]
Save the glossary in Rich Text format for inclusion as part of a larger project document	Generate a Report ^[364]
Create glossary terms and definitions from text in the Notes window, or from any Notes or Description fields that have the Notes toolbar	Notes ^[769]
Insert existing glossary terms into any other Notes fields	Notes ^[769]

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions
- You might include a Glossary Report in your project requirements or functional specifications documents
- You can transport glossary definitions between models, using Export Reference Data and Import Reference Data

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

3.6.3.3.1 Project Glossary Tab

The Project Glossary tab in the Project Information window shows displays the glossary terms already defined for your model.

Access: [View | More Project Tools | Project Information > Project Glossary tab](#) (Alt+2)

Use to:

- Add glossary terms
- Modify glossary terms
- Change a definition type for all terms of that type
- Filter the display show terms of a specific type only
- Print the displayed list of terms
- Delete glossary terms

How to:

Right-click on an entry on the Glossary tab, and select the appropriate context menu option for the operation you intend to perform (as listed above); if you select the:

- **Add New** or **Modify Selected** options, the Glossary Detail dialog displays; complete the fields as indicated in the following table:

Field	Usage	See also
Term	Type the term to include in the glossary	
Type	(Mandatory) Select the required type If you require a different glossary type, click on the (...) (browse) button and specify the name of the new type This field applies the type <i>only</i> to the selected term; you can rename a type for <i>all</i> terms of that type, using the context menu as described below	
Meaning	(Mandatory) Type the definition or description of the term If necessary, format the text of this description using the Notes toolbar at the top of the field	Notes
New	Clear the dialog fields so that you can define a new Glossary term	
Apply	Save the new or updated glossary entry	
Delete	Delete the entry A prompt displays to confirm the deletion; click on the Yes button to remove the term from the glossary	

- **Rename type** option, the Rename Glossary Type dialog displays on which you enter the alternative type name (either one of the existing types or a new type); when you click on **OK**, *all* entries of the original type are changed to the new type
- **Set term filter** option, the Term Type Filter dialog displays on which you enter the type of glossary term to list; when you click on **OK**, only items of that type are shown on the Project Glossary tab
- **Remove term filter** option, the Project Glossary tab refreshes to show glossary items of all types
- **Print List** option, the Print dialog displays on which you define the printing parameters; when you click on **OK**, the currently-displayed list prints out
- **Delete** option, a prompt displays to confirm the deletion; click on the **Yes** button to remove the term from the glossary

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions

3.6.3.3.2 The Glossary Dialog

The Glossary dialog enables you to create and maintain definitions of terms used in your project, as a Glossary.

Access: [Project](#) | [Documentation](#) | [Glossary](#)

Use to:

- Add, modify and delete glossary entries
- Filter the dialog display to list entries of a specific type only

How to:

Select the **Glossary** menu option and complete the Glossary dialog as indicated below

Field	Usage	See also
Glossary Term	Type the term to include in the glossary	
Glossary Type	(Mandatory) Select the required type If you require a different glossary type, click on the (...) (browse) button and specify the name of the new type This field applies the type <i>only</i> to the selected term; you can <i>rename</i> a type for all terms of that type ^[362] , using the Project Glossary tab of the Project Information window	Project Glossary Tab ^[362]
Description	(Mandatory) Type the definition or description of the term If necessary, format the text of this description using the Notes toolbar at the top of the field	Notes ^[770]
Limit Display To	Select the appropriate glossary type to filter the Type Term list to show entries of that type only Select All to display all glossary entries	
New	Clear the dialog fields so that you can define a new Glossary term	
Type Term	Review the list of defined glossary terms; click on an entry to edit it in the dialog fields	
Save	Save the new or updated glossary entry	
Delete	Delete the entry selected from the Type Term list	
Report	Generate and print a glossary report ^[364]	Generate a Report ^[364] Glossary Report Output Sample ^[365]

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions

3.6.3.3.3 Generate a Report

You can generate a rich text format file of your model's glossary, which you can either print or attach to any other appropriate document file.

The file can contain all glossary entries, or just those of selected types.

Access: [Project](#) | [Documentation](#) | [Glossary: Report](#)

Use to:

- Generate a formatted RTF document file of your model glossary

How to:

To generate an RTF report of the project glossary, follow the steps below:

Step	Action	See also
1	Select the Glossary menu option The Glossary dialog displays	
2	Click on the Report button The Glossary Report dialog displays	
3	In the Filename field, type or select a filename for the glossary	
4	In the Heading field, type a suitable heading for the glossary	
5	In the Include Glossary Items panel, select the checkbox for each type of glossary entry to include To select all types of entry, click on the Select All button	
6	If necessary, to define the page setup and language for the report, click on the Page Setup and/or Language buttons	
7	To include page breaks, select the Page break between sections checkbox	
8	Click on the Generate button to generate the report	
9	Click on the View button to open the report	Glossary Report Output Sample ^[365]

3.6.3.3.1 Glossary Report Output Sample

An example of the output from a Glossary report is shown below:

Glossary

Business Terms

Accounting Periods

A defined period of time whereby performance reports can be extracted. (normally 4 week periods).

Customer

A person or a company that requests An entity to transport goods on their behalf.

Technical Terms

Association

A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link.

Component Model

The component model provides a detailed view of the various hardware and software components that make up the proposed system. It shows both where these components reside and how they inter-relate with other components. Component requirements detail what responsibilities a component has to supply functionality or behavior within the system.

Deployment Model

A model of the system as it is physically deployed.

Extends Relationship

A relationship between two Use Cases in which one Use Case 'extends' the behavior of another. Typically this represents optional behavior in a Use Case scenario - for example a user might optionally request a list or report at some point in a performing a business Use Case.

3.6.4 Project Task Allocation

The Project Task Allocation window enables you to review the allocation of work to elements in the project, focussing on either the elements that require work, or the resources required to perform the work. The window primarily shows information that is entered through other windows and dialogs, but once a record exists in the window you can edit it and, for example, add to or change the resources on an element.

Access: [View | Task Allocations](#)

Topics:

Topic	Detail	See also
Resource View	The Resource View shows the current commitments of each allocated resource in the project as a list of allocation records and a GANTT chart of the progress of the allocations	Resource View [366]
Element View	The Element View lists the elements in the project that have resources assigned to them, and the resources assigned to each element, as a list of allocation records and a GANTT chart of the progress of the allocations	Element View [368]
Report View	A Resource report shows how your resources are deployed in your project, displaying a list of all elements that have resources allocated to them	Report View [370]

3.6.4.1 Resource View

The Resource View of the Project Task Allocation window shows the current commitments of each allocated resource in the project as a list of allocation records and a GANTT chart of the progress of the allocations.

The display initially shows the resources and their overall commitment; click on the 'plus-box' to the left of the name of the resource to expand the entry to show the elements and the allocation period for each element.

The display shows both complete allocations and those that are still in progress; an internal filter hides completed allocations two weeks after the end date, and incomplete allocations one month after the end date.

Access: [View | Task Allocations > Resource View](#)

Use to:

- Check the dates on which specific resources or all resources are currently allocated
- Check the elements to which each resource is allocated
- Check the dates on which a resource is allocated to work on a specific element
- Check the general progress of the resource in completing the work
- Check the specific details of the allocation of a resource to an element and task or role

How to:

On the Resource View tab, make selections from the options as indicated below:

Option	Usage	See also
Filter the display	Right-click on the window and select the Apply Filter context menu option The Filter Bar displays at the top of the Resource and Role or Task columns; type in a text string to filter for <i>element names</i> and <i>role or task</i> names, respectively, that begin with the text	List Header ^[458]
Display the properties of the element to which the resource is assigned	Right-click on the entry and select the Object Properties context menu option The Properties dialog for the element displays	Properties Dialog ^[660]
Display the resource allocation details	Right-click on the entry and select the Task Properties context menu option The Assigned Resources dialog displays; you can edit the details and, if necessary, change the resource allocated to the element	Resource Allocation ^[349] The Project Management Window ^[347]
Refresh the report	Right-click on the entry and select the Refresh context menu option The content of the display is refreshed and collapsed to resource level	
Display the Resource Allocation records for the element	Right-click on the entry and select the Locate in Resource Window context menu option The Project Management window opens at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel You can edit the details and, if necessary, change the resource allocated to the element	Resource Allocation ^[349] The Project Management Window ^[347]
Reposition the GANTT chart to	Right-click on the entry and select the required option: <ul style="list-style-type: none"> • Go To Task Start 	

Option	Usage	See also
show the start date or end date of an allocation, or today's date	<ul style="list-style-type: none"> • Go To Task End • Go To Today's Date <p>The display shifts to put the required date in the center of the chart</p>	
Locate the element in the Project Browser	<p>Right-click on the element name and select the Find in Project Browser context menu option</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted</p>	
Save an image of the GANTT chart to file	<p>Right-click on the tab and select the Save Image to File context menu option</p> <p>The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to</p>	
Save an image of the GANTT chart to the clipboard	<p>Right-click on the tab and select the Copy Image to Clipboard context menu option</p> <p>You can paste the image from the clipboard into your preferred graphics package</p>	

3.6.4.2 Element View

The Element View of the Project Task Allocation window lists the elements in the project that have resources assigned to them and the resources assigned to each element, as a list of allocation records and a GANTT chart of the progress of the allocations. An allocation can be partial, in that a role or task is assigned but no specific resource has been identified.

The display initially shows the elements and their overall resource commitment; click on the 'plus-box' to the left of the name of the element to expand the entry to show the resources and the allocation period for each resource.

The display shows both complete allocations and those that are still in progress; an internal filter hides completed allocations two weeks after the end date, and incomplete allocations one month after the end date.

Access: [View | Task Allocations > Element View](#)

Use to:

- Check the dates on which specific elements have resources currently allocated
- Check which resources are currently allocated to an element
- Assign further resources to the element
- Check the dates on which a resource is allocated to work on a specific element
- Check the general progress of the resource in completing the work
- Check the specific details of the allocation of a resource to an element and task or role

How to:

On the Element View tab, make selections from the options as indicated below:

Option	Usage	See also
Filter the display	<p>Right-click on the window and select the Apply Filter context menu option</p> <p>The Filter Bar displays at the top of the Element and Role or Task columns; type in a text string to filter for <i>resource names</i> and <i>role or task</i> names, respectively, that begin with the text</p>	List Header ^[458]
Display the properties of the element	<p>Right-click on the element and select the Properties context menu option</p> <p>The Properties dialog for the element displays</p>	Properties Dialog ^[660]
Display the resource allocation details	<p>Right-click on the resource and select the Properties context menu option</p> <p>The Assigned Resources dialog displays; you can edit the details and, if necessary, change the resource allocated to the element</p>	Resource Allocation ^[349] The Project Management Window ^[347]
Assign a new resource to the element	<p>Right-click on the resource and select the Assign Resource context menu option</p> <p>The Assigned Resources dialog displays; complete this as for the Resource Allocation tab of the Project Management window</p>	Resource Allocation ^[349] The Project Management Window ^[347]
Refresh the report	<p>Right-click on the display and select the Refresh context menu option</p> <p>The content of the display is refreshed and collapsed to element level</p>	
Display the Resource Allocation records for the element	<p>Right-click on the entry and select the Locate in Resource Window context menu option</p> <p>The Project Management window opens at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel</p> <p>You can edit the details and, if necessary, change the resource allocated to the element</p>	Resource Allocation ^[349] The Project Management Window ^[347]
Reposition the GANTT chart to show the start date or end date of an allocation, or today's date	<p>Right-click on the entry and select the required option:</p> <ul style="list-style-type: none"> • Go To Task Start • Go To Task End • Go To Today's Date <p>The display shifts to put the required date in the center of the chart</p>	
Locate the element in the Project Browser	<p>Right-click on the element name and select the Find in Project Browser context menu option</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted</p>	
Save an image of the GANTT chart to file	<p>Right-click on the tab and select the Save Image to File context menu option</p> <p>The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to</p>	

Option	Usage	See also
Save an image of the GANTT chart to the clipboard	<p>Right-click on the tab and select the Copy Image to Clipboard context menu option</p> <p>You can paste the image from the clipboard into your preferred graphics package</p>	

3.6.4.3 Report View

The *Report View* shows how your resources are deployed in your project, displaying:

- a list of all elements that have resources allocated to them, and the type of each element
- the resource allocated, and the role played by that resource
- the start and end dates of the allocation
- the time allocated, expected and expended
- the percentage completion of the allocation

You can tailor the displayed information by:

- Hiding columns of information
- Grouping types of information
- Filtering the data by status
- Filtering the data by start date or end date
- Filtering any column to show only a specific value

Having displayed the information you require, you can print it.

Access: [View | Task Allocations > Report View](#)

Use to:

Identify, for example:

- What elements each resource is working on
- What resources are assigned to an element or package
- How close to completion your resources are, or
- How long a resource is assigned to a particular area.

How to:

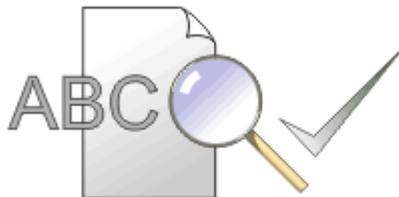
On the Report View tab, make selections from the options as indicated below:

Option	Usage	See also
Run the report	<p>Click on the first icon in the Report View toolbar (the rotating arrows)</p> <p>The report results display</p> <p>If you have the report open for a while, you can update the information; either:</p> <ul style="list-style-type: none"> • Run the report again or • Right-click on the content and select the Refresh context menu option 	
Adjust column	Drag and drop column headings into the sequence you	List Header ^[458]

Option	Usage	See also
headings	<p>require</p> <p>Right-click on the column headings and select the Field Chooser context menu option, to add or remove columns using the Field Chooser dialog</p>	
Group entries by column heading	<p>Right-click on the column headings and select the Enable Group Box context menu option, to cluster the report information according to your preferred hierarchy of column headings</p>	List Header ^[458]
Filter columns	<p>Either:</p> <ul style="list-style-type: none"> • Click on the third toolbar icon from the <i>right</i> (the spy glass), or • Right-click on the column headings and select the Toggle Filter Bar context menu option <p>The filter field displays at the top of every column</p> <p>Type in whatever text string you require in the appropriate column to filter the report to show only entries containing that text string in that column</p>	List Header ^[458]
Filter by degree of completion	<p>In the first field in the toolbar, click on the drop-down arrow and select one of the following values:</p> <ul style="list-style-type: none"> • All - Display all entries regardless of degree of completion • Completed - Display only those entries where the allocation is 100% completed • Above cut-off - Display only those entries that are more than a certain percentage complete • Below cut-off - Display only those entries that are less than a certain percentage complete <p>In the second field, either type a threshold value or increment to the value using the up/down arrows, to set the percentage completion for the Above/Below cut-off options</p>	
Filter according to start/end date	<p>Click on the fourth toolbar icon from the right (the funnel)</p> <p>The Resource Filters dialog displays</p> <p>In the Start Date and/or End Date field, click on the drop-down arrow and select the appropriate qualifier:</p> <ul style="list-style-type: none"> • After • Before • Equal To • Not Equals <p>In the date fields, click on the checkbox to activate the fields and either type in the day, month and year or click on the drop-down arrow to select the date from a calendar</p> <p>The fields have an AND relationship; an entry must satisfy both date criteria before it is displayed</p>	
Display the properties of the element to which the resource is assigned	<p>Right-click on the entry and select the Object Properties context menu option</p> <p>The Properties dialog for the element displays</p>	Properties Dialog ^[660]

Option	Usage	See also
Display the resource allocation details	Right-click on the entry and select the Task Properties context menu option The Assigned Resources dialog displays	Resource Allocation [349] The Project Management Window [347]
Display the Resource Allocation records for the element	Right-click on the entry and select the Locate in Resource Window context menu option The Project Management window opens at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel	Resource Allocation [349] The Project Management Window [347]
Locate the element in the Project Browser	Right-click on the entry and select the Find in Project Browser context menu option The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted	
Print the report	Either: <ul style="list-style-type: none"> Click on the second toolbar icon from the right (the printer) or Right-click on an entry in the report and select the Print context menu option <p>The Print dialog displays, on which you specify the printer to use and the characteristics of the print job</p>	

3.6.5 Spell Checking



Enterprise Architect provides a powerful spell checking facility. This operates at both project level and package level.

Topics:

Topic	Detail	See also
Automatic Spell Checking	The spell checker can be set to run automatically, so that it highlights possible errors in text as it is created or pasted in: <ul style="list-style-type: none"> Select Tools Options > Objects Deselect or select the Disable spelling checkbox 	
Spelling Configuration	You can configure what types of text error the spell checker should detect, using the Spelling Options dialog	Select Spell Checker Options [373]
Using Languages	Enterprise Architect is supplied with dictionaries for US	Use Languages

Topic	Detail	See also
Other Than English	English and British English; you can download a set of dictionaries for other languages from the Sparx Systems website	Other Than English ^[375]
Using the Spell Checker	You can run the spell checker manually on the whole model, or on a selected package	Using the Spell Checker ^[376]
Correcting Text	As the spell check progresses, Enterprise Architect highlights any errors or unknown words; you have several options for responding to these potential spelling errors	Correcting Words ^[377]

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options

Learn More:

- [List of Available Permissions](#) ^[205]

3.6.5.1 Select Spell Checker Options

You can configure the spell checker to ignore or detect specific types of text conformation. Ignoring the conformation means to skip over the text without identifying it as an error.

Access: [Tools](#) | [Spelling Tools](#) | [Spelling Options](#)

How to:

To configure the spell checker, select the options for the checks you require on the Spelling Options dialog, as described below:

Option	Action	See also
Ignore capitalized words	Ignore any words beginning with a capital letter, such as <i>David</i> You might enable this option if the text being checked contains many proper names	
Ignore all-caps words	Ignore any words containing all capital letters, such as <i>RADAR</i> You might enable this option if the text being checked contains many acronyms	
Ignore Words with Numbers	Ignore any words containing embedded digits, such as <i>Win95</i> and <i>Q4</i> You might enable this option if the text being checked contains many code words or other symbols containing digits	
Ignore Words with Mixed Case	Ignore any words containing an unusual mixture of upper- and lower-case letters, such as <i>MicroHouse</i> and <i>CapsLock</i> . You might enable this option if the text being checked contains many variable names or other symbols that use case changes to	

Option	Action	See also
	distinguish words	
Ignore Domain Names	Ignore any words that appear to be Internet domain names, such as <i>wintertree-software.com</i>	
Report Doubled Words	Detect any word appearing twice in a row, such as <i>the the</i>	
Case Sensitive	Distinguish between capitalized and non-capitalized words For example, <i>canada</i> would be considered as different from <i>Canada</i> , and therefore reported as a misspelling When the option is disabled , <i>canada</i> and <i>Canada</i> are considered to be identical Note that the performance of the spell checker is reduced if this option is disabled	
Phonetic Suggestions	Suggest alternative words based on phonetic (sounds-like) similarity to the misspelled word; this option tends to improve the correction of badly misspelled words Enabling this option increases the time required to locate suggestions Note that either this option or the Typographical Suggestions option must be enabled in order to list suggestions of alternative words	
Typographical Suggestions	Suggest alternative words based on typographical (looks-like) similarity to the misspelled word. This option is appropriate for people who are generally good spellers Note that either this option or the Phonetic Suggestions option must be enabled in order to list suggestions of alternative word	
Suggest Split Words	Suggest two separate words as a replacement for a misspelling containing two joined words For example, <i>is the</i> would be suggested as a replacement for <i>isthe</i>	
Auto Correct	Automatically change words marked with <i>Auto Change</i> actions to their specified replacements When this option is disabled , a confirmation prompt displays before the words are changed	
Main Dictionary Language	Set the language of the main dictionary used to check spelling; the drop-down list shows only languages for which dictionaries are installed on your system To check spelling in a different language, select the language in the list	Use Languages Other Than English ³⁷⁵
Suggestions	Determine the speed and accuracy of the initial automatic search for suggested replacements for misspelled words Click on the appropriate radio button for your choice	
OK	Close the Options dialog, and save any changes made to the option settings	

Option	Action	See also
Cancel	Close the Options dialog, and discard any changes made to the option settings	

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options

Learn More:

- [List of Available Permissions](#) ^[205]

3.6.5.2 Use Languages Other Than English

Enterprise Architect is supplied with three dictionaries, for US English, Canadian English and British English. Additional dictionaries are available as a set, for download from the registered pages of the Sparx Systems website.

Use to:

- Download and install the language pack of dictionaries so that you can select another language in which to perform the spell check

How to:

To download the additional language dictionary pack, follow the steps below:

Step	Action	See also
1	Access the registered user page on the Sparx Systems web site	Registered User page
2	At the end of the page, download the <i>EADict.zip</i> file from the Enterprise Architect Dictionary section	
3	Unzip the file into the Enterprise Architect install directory - <i>C:\Program Files\Sparx Systems\EA</i> This makes the non-English spelling dictionaries available to the Enterprise Architect spell checker	Select Spell Checker Options ^[373]

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options
- Enterprise Architect .EAP files default to use Jet 3.5 as the database engine; this does not support unicode character sets, which makes the use of certain languages - such as Polish - difficult, including spell checking words in those languages

If you use such languages, it is recommended that you either upsize to a DBMS repository or set JET 4.0 as the database engine

If your .EAP project is not in a Jet 4.0 database, you should also download a copy of the Jet 4.0 EABase

model from the Sparx Systems website, and do an EAP to EAP transfer of your model into the Jet 4.0 file

Learn More:

- [Server Based Repositories](#) ^[148]
- [General](#) ^[422]
- [EABase Jet4 page of the website](#)
- [Perform a Project Data Transfer](#) ^[343]
- [List of Available Permissions](#) ^[205]

3.6.5.3 Using the Spell Checker

Enterprise Architect has an inbuilt spell checker, which you can configure to detect a range of types of possible spelling error in your project, and to use one of several language dictionaries other than English.

Access: **Tools | Spelling Tools | Spell Check Project**
Tools | Spelling Tools | Spell Check Current Package

How to:

To run the spell checker, follow the steps below:

Step	Action	See also
1	Select the Spell Check Project or Spell Check Current Package menu option, as required The Spell Check dialog displays	
2	(Spell Check Project option) Select the checkbox against each of the items to spell check within your model	
3	Click on the Start button to begin the spell check As the spell check proceeds, the text being checked displays in the text panel at the bottom of the screen; if an error is detected, the Check Spelling dialog displays, offering several options to correct the error	Correcting Words ^[377]

Notes:

- The **Spell Check Project** menu option enables you to check spelling for the entire project
- The **Spell Check Current Package** option only checks the package currently open, and does not enable you to select items to check
- Enterprise Architect currently supports checking an entire model or a single package; a future release should support more detailed spell checking at the element and diagram level
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options

Learn More:

- [List of Permissions](#) ^[205]

3.6.5.3.1 Correcting Words

As a spell check progresses, Enterprise Architect highlights any errors or unknown words in the Check Spelling dialog. The inbuilt spell check stores user-defined words in the user dictionary (*userdict.txt*) stored in the Enterprise Architect installation directory. During the spell check process, if you add a word, it is written into this file for later reference.

Use to:

- Correct the spelling of a word
- Ignore the error
- Add the word to the user dictionary
- Suggest alternatives or
- Otherwise assist in the spelling correction process

How to:

As the Spell Checker identifies possibly mis-spelled words, make the appropriate response:

Action	See also
<ul style="list-style-type: none"> • Modify the spelling by hand and click on the Change or Change All button to change the word to that spelling • Click on a suggested alternative and click on the Change or Change All button to change the word to that spelling • Click on the Ignore or Ignore All button to exclude the word from the spell check • Click on the Add button to add the word to the user dictionary • Click on the Suggest button to list alternative spellings or words • Click on the Cancel button to abort the spell check entirely 	

3.6.6 Personal Information

The Personal Information window enables you to record and manage your personal work within the project. The window operates on your identity as a defined Author on the project.

Access: [View | Personal Information](#)

Topics:

Topic	Detail	See also
Model Mail	You have the facility to send simple emails to registered project team members, and to receive messages from your team members	Internal Mail ^[378]
Allocated Work	Presents a GANTT chart on which you record the work that you are currently engaged in.	Recording Work ^[382]
Project Tasks	Enables you to monitor and maintain the work tasks that have been assigned to you, or that you have created yourself	Monitor Your Tasks ^[385]
Workflow	Enables you to list and run your workflow scripts	Monitor Workflow ^[386]

Topic	Detail	See also
Working Sets	Enables you to capture your current work area (diagrams and facilities currently open) as a working set for subsequent re-use	Working Sets ^[387]

3.6.6.1 Internal Mail

The *Model Mail* tab shows all emails that you have received from project team members within the project, under your User Security ID. You can open and respond to these emails and create new ones.

The *Sent Items* tab shows all the emails you have sent to project team members under your User Security ID. You can forward these emails and make further responses (see the final table in this topic).

This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with User Security enabled; the mail system uses the individual and group users defined in User Security.

Access: [View | Personal Information > Model Mail](#)
[View | Personal Information > Sent Items](#)

Use to:

- Read messages from other project team members
- Read and respond to messages sent to you as a member of a group that holds a group mailbox
- Organize and sort both sent and received messages
- Create new messages, and reply to and forward received messages
- Delete messages from your In tray and Sent Items mailboxes

How to:

On the Model Mail tab, review your messages as indicated below:

Option	Usage	See also
Open a message	Double-click on the message If the message contains a hyperlink, click on the link to open or display the target file or object	
Flag	Check the flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team)	
Status	Check for the read or unread icon; unread message items are anyway displayed in bold	
Sender	The name of the project team member who sent the message	
Subject	The topic of the message	
Date	Indicates the age of the message, whether it was sent: <ul style="list-style-type: none"> • Today • Yesterday • This week • This Month • Last Month • Older 	

Option	Usage	See also
Sent	The exact date and time the message was sent	
Select Columns	<p>Right-click on the column headings and select the Field Chooser menu option, which enables you to add or remove specific columns from the display</p> <p>You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence</p> <p>Select the To column to display the names of the people to whom the email was sent; this is very useful to identify whether the email was sent to you as an individual or as a member of a group - if you received the email as a member of a group, remember that your actions on this single instance of the email affect all other recipients</p>	List Header [458]
Reorganize messages in the list	<p>Either:</p> <ul style="list-style-type: none"> • Click on the column heading and click on the arrow head to list items in order or reverse order, or • Right-click on the column head and select the Enable Group Box option to organize the messages into groups 	

On the Model Mail tab, make selections from the toolbar options as indicated below:

Option	Usage	See also
Compose Message	Open the Model Message dialog, through which you create and send a mail message	Create a Message [38]
Delete Messages	Delete a selected message or messages; you are prompted to confirm the deletion	
Reply	Open the Model Message dialog, through which you create a response to the sender of the message, which includes a copy of the message and any earlier messages in the thread	Create a Message [38]
Reply to All	Open the Model Message dialog, through which you create a response to the sender of the message and the other recipients, which includes a copy of the message and any earlier messages in the thread	
Forward	Open the Model Message dialog, through which you forward the message to other project team members with, if necessary, your own comments	
Mark as Unread	Mark the selected messages as unread	
Mark as Read	Mark the selected messages as read	
Viewing Panel	<p>Select to:</p> <ul style="list-style-type: none"> • Display the message text viewing panel underneath or to the right of the message list, or • Hide it so that you have to open the message to read it 	
Help Contents	Display this Help topic	

On the Model Mail tab, make selections from the context menu options as indicated below:

Option	Usage	Shortcut	See also
Compose New Message	Click to open the Model Message dialog, through which you create and send a mail message		
Reply	Create a response to the sender of the message, which includes a copy of the message and any earlier messages in the thread	Ctrl + R	Create a Message [38]
Reply to All	Create a response to the sender of the message and the other recipients, which includes a copy of the message and any earlier messages in the thread	Ctrl + Shift + R	
Forward	Forward the message to other project team members with, if necessary, your own comments	Ctrl + F	
Set Message Flag	Select the appropriate flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team)		
Mark as Unread	Mark the selected messages as unread	Ctrl + U	
Mark as Read	Mark the selected messages as read	Ctrl + Q	
Delete	Delete a selected message or messages; you are prompted to confirm the deletion		

On the Sent Items tab, manage the messages you have sent as indicated below:

Option	Usage	See also
Open a message	Double-click on the message From the open message you can reply to the sender, reply to the sender and all other recipients, or forward the message to other team members	Create a Message [38]
Flag	Check the flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team)	
To	Check the name(s) of the project team member(s) to whom the message was sent	
Subject	The topic of the message	
Date	Indicates the age of the message, whether it was sent: <ul style="list-style-type: none"> • Today • Yesterday • This week • Last Week • This Month • Last Month • Older 	
Sent	The exact date and time the message was sent	
Select Columns	Right-click on the column headings and select the Field Chooser menu option, which enables you to add or remove specific columns from the display	

Option	Usage	See also
	<p>You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence</p> <p>Add the Sender column to your window; this is useful to identify whether you have responded to an email as an individual or as a member of a user group</p>	List Header <small>458</small>
Reorganize messages in the list	<p>Either:</p> <ul style="list-style-type: none"> Click on the column heading and click on the arrow head to list items in order or reverse order, or Right-click on the column head and select the Enable Group Box option to organize the messages into groups 	
Delete messages	<p>Right-click on the message and select the Delete context menu option</p> <p>You are prompted to confirm the deletion</p>	
Reply to All	<p>Right-click on the message and select the Reply to All context menu option</p> <p>The Model Message dialog displays</p>	Create a Message <small>381</small>
Forward	<p>Right-click on the message and select the Forward context menu option</p> <p>The Model Message dialog displays</p>	

3.6.6.1.1 Create a Message

The *Model Message* dialog enables you to compose messages to project team members within the project, under your Author ID.

Access: [View | Personal Information > Model Mail](#)

Use to:

- Create messages to other project team members
- Reply to messages from other team members
- Forward messages to other team members
- Link model components to the message, or add hyperlinks to files or other objects of relevance

How to:

To create a message to another project team member as a new message, a reply to a message, or a forwarded message, follow the steps below:

Step	Action	See also
1	If your message or response concerns a diagram or model object, you could click on that object in the Project Browser now so that it is immediately available to be linked to your message	
2	<p>Select the appropriate toolbar icon, context menu option or message button to create, reply to or forward a message</p> <p>The Model Message dialog displays</p>	Internal Mail <small>378</small>
3	If you are creating or forwarding a message, or you want to send a reply to a wider	

Step	Action	See also
	audience, for <i>each</i> person you intend to send the message to: <ul style="list-style-type: none"> Click on the To: button A list of project team member and group IDs displays Scroll to the required name and double-click on it The name is added to the To: field 	
4	In the Subject field, type or edit the subject of the message as necessary	
5	In the Flag field click on the drop-down arrow and on the appropriate flag color or option for your message	
6	In the text panel, write the text of your message You can format the text using the facilities of the Notes toolbar at the top of the field	Notes Toolbar ^[770]
7	If you intend to link a diagram or model object to this message, place the cursor at the appropriate position in your message text and click on the Insert Quick Link button Select the object you highlighted in Step 1; a hyperlink to that object is inserted in the message text	
8	If you did not select a model object and now intend to link to one, or you want to link to another target such as a file, Model Search or Team Review message, click on either: <ul style="list-style-type: none"> The Hyperlink icon in the toolbar or The Insert Quick Link button and select the Other option The Hyperlink Details dialog displays; create the required link	Hyperlinks ^[1292]
9	Click on the Send button to send the message to the recipients	

Notes:

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with User Security enabled

3.6.6.2 Recording Work

The Allocated Work tab lists the elements to which *your* model Author ID has been allocated as a resource, where your Author ID is the same as:

- Your Enterprise Architect security user ID, if security has been enabled, or
- Your workstation login ID if security has not been enabled

For each element, the tab:

- In the left-hand panel lists the roles or tasks assigned to you as a resource on that element, for which the **Complete %** field value is less than **100**
- In the right hand panel displays a GANTT chart showing your progress in performing each role or task

You can add further work items for an element through the Allocated Work tab; however, you cannot delete any records. A record is no longer listed when the **Complete %** field value is **100**.

You or your supervisors can also add records through the Resource Allocation tab of the Project Management window.

Access: [View | Personal Information > Allocated Work](#)

Use to:

- View the elements to which you have been allocated as a resource, and the tasks you have been set to complete
- View and edit the properties of the elements you have been allocated to as a resource
- View and edit the details of each assigned role or task including, if you follow your project team procedures, re-assigning your task to another resource
- Add further tasks to an element
- Filter the display to show only items having a specific text element in the title, or restrict selected items to show just the element, not the tasks
- Locate the elements in the Project Browser
- Locate the tasks in the Resource Allocation tab of the Project Management window
- Print a list of tasks
- Capture an image of the display to a file or to the clipboard

How to:

On the Allocated Work tab, make selections from the options as indicated below:

Option	Usage	See also
Review element properties	<p>Double-click on the element name</p> <p>(Alternatively, right-click on the element name and select the Properties context menu option)</p> <p>The Properties dialog for the element displays; review the pages as required</p>	Properties Dialog <small>[660]</small>
Review resource task details	<p>Double-click on the task item</p> <p>(Alternatively, either:</p> <ul style="list-style-type: none"> • Double-click on the progress bar for the item on the GANTT chart, or • Right-click on the task or role name and select the Properties context menu option) <p>The Assigned Resources dialog displays, which has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode</p> <p>Should it be necessary to reassign the item to another resource, click on the drop-down arrow on the Resource field and select the appropriate Author ID; when you save the changes, the item no longer appears in the list of tasks assigned to you</p>	Resource Allocation <small>[349]</small> The Project Management Window <small>[347]</small>
Create new task item	<p>Right-click on the element name and select the Assign Resource context menu option</p> <p>The Assigned Resources dialog displays, with your Author ID in the Resource field</p> <p>Complete the dialog as for the Resource Allocation tab of the Project Manager window</p>	Resource Allocation <small>[349]</small>
Refresh display to incorporate changes	<p>Your work item can be edited in a number of places in Enterprise Architect, such as the Project Task Allocation window and the Resource Allocation tab; the element to which it is assigned can be edited in these and many other areas</p>	Resource Allocation <small>[349]</small> Project Task Allocation <small>[366]</small>

Option	Usage	See also
	To refresh the display with any changes made elsewhere, right-click on the display and select the Refresh context menu option	Properties Dialog [660]
Display the Resource Allocation tab of the Project Management window	Right click on the list of items and select the Show Resources Window context menu option The Resource Allocation tab of the Project Management window displays, listing all items for all resources associated with the element, and displaying the details for the first item in the list	Resource Allocation [349]
Locate the item in the Resource Allocation tab of the Project Management window	Right-click on the work item and select the Locate in Resource Window context menu option The Resource Allocation tab of the Project Management window displays, showing the item details for the selected item only	Resource Allocation [349]
Locate the element in the Project Browser	Right-click on either the element or the work item and select the Find in Project Browser context menu option The appropriate package hierarchy expands in the Project Browser, and the selected element is highlighted	
Expose hidden sections of the work item progress	Some items might cover a long period of time, and you might not be able to display the complete progress line for the item in the GANTT chart To locate the start point, expected end point, or today's date on the progress line for an item, right-click on either the item or the progress line and select the appropriate context menu option: <ul style="list-style-type: none"> • Go To Task Start • Go To Task End • Go To Today's Date The GANTT chart shifts left or right to position the required point in the center of the display	
Filter the work items	You can refine the list of work items to show only those containing text that matches the filter item Right-click on the tab and select the Apply Filter context menu option The filter bar displays at the top of the panel; type in the filter text As you type, the items listed and the GANTT chart are filtered to show only items where the item names match the text string The filter does not operate on the element names If you do not want to use the filter, right-click on the tab and select the Hide Filter Bar context menu option	
Capture an image of the Allocated Work data as a graphics file	Right-click on the tab and select the Save Image to File context menu option The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to	

Option	Usage	See also
Copy an image of the Allocated Work data to the clipboard	<p>Right-click on the tab and select the Copy Image to Clipboard context menu option</p> <p>You can paste the image from the clipboard into your preferred graphics package</p>	

3.6.6.3 Monitor Your Tasks

The Project Tasks tab lists the tasks that you either:

- Own, or
- Are assigned to

These tasks can be created through the:

- Project Tasks tab itself
- Tasks tab of the Project Information window
- Project Tasks view of the Resource Calendar

Access: [View | Personal Information > Project Tasks](#)

Use to:

- Review the status and progress of tasks that you are responsible for
- Create new tasks
- Update existing tasks
- Delete tasks
- Modify and filter the display of task information

How to:

On the Project Tasks tab, make selections from the options as indicated below:

Option	Usage	See also
Add or Modify a task	You add or modify a task through the Task Detail dialog, which displays when you double-click on an entry (edit) or blank line (create)	Add, Modify and Delete Tasks ^[357]
Delete a task	<p>Right-click on the message and select the Delete context menu option</p> <p>You are prompted to confirm the deletion</p>	
Select columns	<p>Right-click on the column headings and select the Field Chooser menu option, which enables you to add or remove specific columns from the display</p> <p>You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence</p>	List Header ^[458]
Reorganize tasks in the list	<p>Either:</p> <ul style="list-style-type: none"> • Click on the column heading and click on the arrow head to list items in order or reverse order, or • Right-click on the column headings and select the 	List Header ^[458]

Option	Usage	See also
	<p>Enable Group Box option to organize the messages into groups</p> <p>You can also use the filter bar to filter the display on an appropriate column value, such as the value New in the Status column; to display or hide the filter bar, right-click on the column headings and select the Toggle Filter Bar context menu option</p>	
<p>Set persistent Status filter</p>	<p>Right click on the tab and select the Set term filter context menu option</p> <p>The Filter by Status dialog displays, which enables you to select to list tasks of any status or only of one specific status</p> <p>The filter you set persists when you close the Personal Information window or exit from Enterprise Architect</p>	
<p>Print the task list</p>	<p>Right click on the tab and select the Print List context menu option</p> <p>The Print dialog displays, on which you specify the local printer and the print characteristics</p>	

Learn More:

- [Project Task Display](#)^[398]
- [Project Tasks](#)^[358]

3.6.6.4 Monitor Workflow

The Workflow tab lists all of the workflow scripts that you have created:

You can execute any of the scripts listed on the tab.

Access: [View | Personal Information > Workflow](#)

Use to:

- Select and execute your workflow scripts

How to:

On the Workflow tab, double-click on the required script to execute it.

Notes:

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

Learn More:

- [Workflow Scripts](#)^[237]

3.6.6.5 Working Sets

The Working Sets tab lists the working sets that you have created.

A working set, rather like a project shortcut, is a group of diagrams and views that you have either captured from a work session or specifically drawn together, in either case using the Create Working Set dialog. It is a personal list, not visible to or active for any other user.

A *view* is:

- A Relationship Matrix profile to open
- The Team Review
- A Model Search to perform

Access: [View | Personal Information > Working Sets](#)

Use to:

- Capture your currently-open diagrams and views to resume work in the same configuration in a subsequent work session
- Create a working set by selecting diagrams, searches and profiles to open
- Open an existing working set to resume work in the required configuration (manually and automatically)
- Change a working set to remove, replace or add to the diagrams and views defined in it
- Delete a working set

How to:

On the Working Sets tab, select the required option as indicated below:

Option	Usage	See also
Open Working Set	<p>Click on a working set name and either:</p> <ul style="list-style-type: none"> • Select the third icon in the toolbar or • Right-click and select the Open Working Set context menu option <p>Enterprise Architect opens all the diagrams and views listed in that set</p> <p>If you already have a working set open, the 'new' set of views is added to the original set, with the last-listed diagram or view on top and in focus</p> <p>You can open only one Model Search, Matrix profile or Team Review at a time; therefore, the last-opened Search, profile or Team Review replaces the previous instances</p>	
Create New Working Set	<p>Either:</p> <ul style="list-style-type: none"> • Click on the first icon in the toolbar or • Right-click on the body of the tab and select the Create Working Set context menu option <p>The Create Working Set dialog displays</p> <p>In the Name field, type a name for the working set</p> <p>If you have diagrams and views <i>open</i>, the dialog lists them in the order in which they were opened; you can change the order using the '</p>	

Option	Usage	See also
	<p>hand buttons at the bottom of the dialog</p> <p>Either select the checkbox against each diagram or view to include in the working set, or click on the Include All button</p> <p>To add more diagrams or views that are not currently open, click on the Add Other button and select an option from the list:</p> <ul style="list-style-type: none"> • Add Diagram • Add Matrix Profile • Add Search • Add Team Review <p>You are prompted to select the required diagram, profile, search or Team Review topic; the view is then added to the list with its checkbox selected</p> <p>Click on the OK button to create the working set</p>	
<p>Build a Working Set from open diagrams or files</p>	<p>A variation on creating a working set is to generate the set from diagrams or files that are currently open</p> <p>Right-click on the body of the tab and select the Create Working Set From context menu option, and either the Open Diagrams or Open Files sub-option as appropriate</p> <p>The Create Working Set dialog displays</p> <p>In the Name field, type a name for the working set</p> <p>The dialog lists the diagrams and views that are currently open, in the order in which they were opened; with the checkboxes against the open diagrams or the open files already selected</p> <p>You can either:</p> <ul style="list-style-type: none"> • Immediately click on the OK button to create the working set from the current environment, or • Change the content and selections as described for <i>Create New Working Set</i> (above) 	
<p>Add an open view or diagram to a working set</p>	<p>If, in your work, you open a diagram or view that you decide to add to a working set:</p> <ol style="list-style-type: none"> 1. Open the Working Sets tab 2. Right-click on the required set name and select the Add Active View context menu option <p>The open view is added to the working set</p>	
<p>Duplicate a Working Set</p>	<p>If you want to use a variation of a working set without losing the original, you can create a copy and amend that copy as required</p> <p>Right-click on the source working set and select the Copy as New context menu option</p> <p>The Create Working Set dialog displays</p> <p>In the Name field, type a name for the working set</p> <p>Each of the listed diagrams and views is already selected; make any changes you require, and click on the OK button to save the list as a new working set</p>	
<p>Edit a working set</p>	<p>To change the diagrams and views in a working set, or change the set name, click on the set name and either:</p>	

Option	Usage	See also
	<ul style="list-style-type: none"> • Click on the second icon in the toolbar or • Right-click and select the Edit context menu option <p>The Create Working Set dialog displays</p> <p>Make any changes you require, and click on the OK button to save the changes</p>	
Delete Working Set, or View in Working Set	<p>Expand the working set and select a specific item (to remove that item) or the set name (to move the whole set)</p> <p>Either:</p> <ul style="list-style-type: none"> • Click on the fifth icon in the toolbar or • Right-click on the set and select the Delete Working Set Item context menu option <p>A prompt displays to confirm the deletion; click on the Yes button</p> <p>The item or set is removed</p> <p>If you delete a working set that you have opened, the diagrams and views from that set remain open until you specifically close each one or close Enterprise Architect</p>	
Automatically apply the working set when the model is opened	<p>Right-click on the working set name and select the Apply when model opens context menu option</p> <p>This option is a toggle that turns the same option off for the previous working set on which it was selected</p> <p>The folder turns green, indicating that the option is on for this working set</p> <p>When you close the model and subsequently open it, the diagrams and views in the working set are automatically opened</p>	
Track working set/ views last opened	<p>You might use several working sets in a work session, and close some views and open others</p> <p>To capture the diagrams and views you actually have open when you close down Enterprise Architect, right click on the body of the tab and select the Store Main Tab View History context menu option</p> <p>When you reopen the model in a new work session, the working set at the top of the tab - <i>Windows open when Model was last closed</i> - is populated with the list of open diagrams and views; this is a read-only set that cannot be edited</p> <p>You can expand the set and double-click on specific windows to re-open them, or right-click on the set and use the Open Working Set option to reopen all the diagrams and views you had open before</p> <p>However, it is recommended that you use the Apply when model opens context menu option on the set; thereafter, whenever you open the model, all the diagrams and views you had open when you closed the model are automatically re-opened</p>	
Locate item in Project Browser	<p>To find a diagram from your working list in the Project Browser, right-click on the diagram name and select the Find in Project Browser context menu option</p> <p>The section of the Project Browser containing the diagram is expanded, and the diagram name is highlighted</p>	

Learn More:

- [Create Project Shortcut](#)^[141]
- [Capture Current Work Environment](#)^[142]

3.6.7 Project Calendar

The Project Calendar provides a project management overview in calendar format of the deployment of resources, timeframes for tasks, and upcoming project events such as meetings and milestones. The calendar displays in the Diagram View workspace, consisting of a:

- Calendar panel, displaying months of the year
- Diary panel sectioned into days or weeks
- Toolbar at the top of the window, which enables you to define what information is displayed and how it is formatted

Access: [View | Project Calendar](#)

Topic	Detail	See also
Calendar Panel	<p>The calendar panel, when viewed normally on the screen, displays a calendar of six months that by default includes the current month</p> <p>By closing or reducing other panels on the screen, and depending on the size of your screen, you can show the months for a period of up to 3 years and six months</p> <p>You can review data for up to several decades into the past, if such data has been recorded, and plan ahead for up to several decades into the future; to display future or past months, either:</p> <ul style="list-style-type: none"> • Click on the arrows in the names of the months on the left and right of the top row of months, or • Click and hold the mouse on the name of a month so that a list of months and years displays, and move the mouse forwards to scroll up the list or backwards to scroll down the list; when you reach the required month, ensure that it is highlighted and release the mouse button <p>On the calendar, today's date has a red border; if you have scrolled the calendar so that today's date is not shown, click on the second icon in the toolbar () to position the current month in the top left of the calendar with today's date highlighted</p>	
Diary Panel	<p>The diary panel displays a set of day fields, each of which can contain one or more markers for an event or task</p> <p>The content of the display is determined by the selections you make in the toolbar, and principally by the object type you select from the first drop-down field:</p> <ul style="list-style-type: none"> • Project Calendar • Allocated Resources • Project Tasks <p>The structure of the display depends on the period you want to review, specified using the icons on the toolbar:</p>	<p>Calendar^[392]</p> <p>Allocated Resources^[396]</p> <p>Project Tasks^[356]</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> • 1 day (Project Calendar only) • 5 days (Mon - Fri) (Project Calendar only) • 7 days (Mon - Sun) • 31 days <p>However, in the Calendar panel you can highlight specific periods of between 1 and 7 days (Project Calendar only) and 14, 21, 28, 35 and 42 days, to display just those periods in the Diary panel</p> <p>If you select a period of between 1 and 7 days, each day is divided into one hour time intervals; you can adjust the time intervals for these periods, using the toolbar (see below)</p> <p>For the current day, the current time is also highlighted</p> <p>Also, if you click on today's date in the Calendar panel the Diary panel changes to show the shortest period (7 days or 1 day) for the object type, again with adjustable time intervals for the times of day</p> <p>For displays of 7 or more days, you can scroll up or down to display the information for earlier or later dates; if you scroll away from today's date, you can return to it by right-clicking anywhere on the display and selecting the Show Today context menu option (not Project Calendar)</p>	
<p>Toolbar</p>	<p>The toolbar options modify what is displayed on the Diary panel, and enable you to add certain kinds of information</p> <p>The icons and fields in the toolbar, from left to right, have the following functions:</p> <ul style="list-style-type: none"> • Add New <object> - enables you to create a new record for an event or task • Assign Resource - for Allocated Resources, enables you to assign a new resource to the task • Show Today - re-focuses the Diary panel and Calendar panel on today's date, for any object type • 1, 5, 7, 31 - sets the number of days displayed in the Diary panel • First drop-down - identifies the type of information to display: Calendar events, allocated Resources, or Tasks • Second drop-down - identifies the type or characteristic of object to filter for (and depends on the value selected in the first field) • Third drop-down - for Allocated Resources, identifies the element type to filter for • Options: <ul style="list-style-type: none"> • Configure Event Subtypes - Displays the Configure Subtype dialog, which you use to define categories of event to record on the Calendar • Time Scale - where the display includes times of day, enables you to reset the time interval • Show End Time - for the Project Calendar, where the times of day are NOT shown and an event occurs within one day, this shows or hides the time at which the event finishes (the start time 	<p>Configure Event Subtypes ³⁹⁸</p>

Topic	Detail	See also
	<p>displays automatically)</p> <ul style="list-style-type: none"> • Show Time As Clocks - toggle between showing start and end times in digital format and as a clock face • Compress Weekend Days - in 31-day format, toggle between showing Saturday and Sunday as separate fields and a pair of half-fields • Show ToolTips - toggle between showing and hiding mouse roll-over tooltips 	

Notes:

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them

3.6.7.1 Calendar

Project Calendar is a display mode of the project Calendar facility. In Project Calendar mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic) containing flags for each *event* that takes place in or through that period. These events can be, for example, meetings, staff absences or commitments, product releases or public holidays.

You can customize the range of events that you record, using the Configure Subtype dialog.

Access: [View | Project Calendar > Project Calendar](#)

How to:

To record an event in the Project Calendar, follow the steps below:

Step	Action	See also
1	Double click on the required day, or the first day in a longer period The Project Event dialog displays	
2	In the Subject field, type the name of the event	
3	If the event is to take place at a specific place, in the Location field type the name of the place or room	
4	In the Category field, click on the drop-down arrow and select the appropriate categorization: <ul style="list-style-type: none"> • Event (such as an external trade show, or internal presentation) • Meeting (whether internal or external) • Milestone/Objective (such as a product test or release) • People (an absence or commitment of a specific staff member) 	
5	If the event is likely to occupy one complete day, select the All day event checkbox; this: <ul style="list-style-type: none"> • Places the event icon at the top of the day field in the Diary, above any time sections • Hides the time sections of the Start time and End time fields 	

Step	Action	See also
6	<p>In the Start time and End time fields, specify the start and end dates and/or times of the event</p> <p>In the date section of each field, either:</p> <ul style="list-style-type: none"> • Click on the drop-down arrow to display the current month and select the month and date, or • Click on the 'spin' arrows to roll the date forwards or backwards by one day at a time <p>In the time section of each field, click on the hour or minute components (which are separate) and either:</p> <ul style="list-style-type: none"> • Type in the required time or • Click on the 'spin' arrows to roll the time backwards or forwards 	
7	<p>In the Event Type field, click on the drop-down arrow and select an appropriate type label for the event; the options change depending on the value you selected for the Category field</p> <p>This provides the fill color for the event icon, as indicated by the fill box next to each option</p>	
8	<p>In the Defined as field, click on the drop-down arrow and select the indicator for whether the event is internal or external to your organization</p>	
9	<p>If the event consists of a telephone call, select the Phone Call checkbox</p>	
10	<p>In the description field, type any notes required on the event; these display in the mouse roll-over tool tip for the event on the Diary panel</p>	
11	<p>If this event repeats at regular intervals, click on the Recurrence button and complete the Event Recurrence dialog (see below)</p>	
12	<p>Click on the OK button to save the event and display it on the Calendar</p>	

To define a recurring event on the Event Recurrence dialog, follow the steps below (some information defaults from the Project Event dialog, but you might have to edit it to accurately define the recurrence details):

Step	Action	See also
1	<p>In the Start and End fields, specify the start and end times of the event; click on the hour or minute components (which are separate) and either:</p> <ul style="list-style-type: none"> • Type in the required time or • Click on the 'spin' arrows to roll the time backwards or forwards 	
2	<p>In the Duration field, click on each of the day(s), hours and minutes portions and either type or 'spin' to the required value</p> <p>The hours and minutes portions are linked to the Start and End fields, so that:</p> <ul style="list-style-type: none"> • As you edit the Start field or the Duration field, the End field adjusts to maintain the duration relative to the start time • As you edit the End field, the Duration field changes to match the difference between the start and end times 	
3	<p>In the Recurrence pattern panel, select the appropriate radio button for the frequency of</p>	

Step	Action	See also
	<p>the event</p> <p>As you select the radio button, the fields on the right of the panel change to further define when the event recurs</p>	
4	<p>Select the appropriate radio buttons and field values, with the following guidance:</p> <ul style="list-style-type: none"> • <i>Daily</i> can be every 1, 2, 3, 4, 5 or 6 days, or every day of the working week • <i>Weekly</i> is on one or more specific days of the week, the events being separated by a period of between 1 and 51 complete weeks • <i>Monthly</i> can be on a specific date or a specific day of the month, the events being separated by a period of between 1 and 11 complete months • <i>Yearly</i> can be on a specific date or a specific day of a specific month 	
5	<p>In the Range of recurrence panel, in the Start field, specify the date on which the event cycle begins</p> <p>Click on the day, month and year components (which are separate) and either:</p> <ul style="list-style-type: none"> • Type in the required value or • Click on the 'spin' arrows to roll the date backwards or forwards 	
6	<p>Select the appropriate radio buttons to indicate that the recurrence cycle:</p> <ul style="list-style-type: none"> • Has no defined end point • Ends after a specific number of occurrences of the event (type in the number) • Ends by a specific date (enter the date) 	
7	<p>Click on the OK button, and again on the Project Event dialog</p> <p>The event icon displays at all points in the Calendar where it is scheduled to occur, as defined in the Event Recurrence dialog; the recurrence is indicated by a graphic depicting two recirculating arrows</p>	
8	<p>If the recurrence is no longer required:</p> <ul style="list-style-type: none"> • Double-click on the icon for any occurrence of the event, to display the Project Event dialog • Click on the Recurrence button to display the Event Recurrence dialog • Click on the Remove Recurrence button; the Event Recurrence dialog closes • Click on the OK button; the Project Event dialog closes <p>Only the first occurrence of the event remains in the Calendar</p>	

Notes:

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the above editions, if security is enabled you must have **Manage Project Calendar** permission in order to create, update or delete Project Calendar events; if security is not enabled, you can change data without this permission
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them

Learn More:

- [Project Calendar](#)^[390]
- [Configure Event Subtypes](#)^[395]
- [List of Available Permissions](#)^[205]

3.6.7.1.1 Configure Event Subtypes

The Configure Subtype dialog enables you to define additional types of event to record on the Project Calendar.

Access: [View](#) | [Project Calendar: Toolbar Options](#) | [Configure Event Subtypes](#)

Use to:

- Define a new event type within a select event category
- Change an existing event type
- Delete an existing event type

How to:

To manage the event types for your project, follow the steps below:

Step	Action	See also
1	On the Configure Subtype dialog, in the Category field, click on the drop-down arrow and select the appropriate category for the event	
2	Click on the Type field and type a suitable name for the event type (or select an existing event from the list below the Category field)	
3	In the Color field, click on the drop-down arrow and select or define a color for the event icon	
4	Click on the Save button to save the new or edited event type to the list for the specified category Click on the New button if you are going to create another event type	
5	If the event type is no longer required, click on the Delete button	
6	If required, you can change the sequence of the event types in the displayed list, using the ' Up Hand ' and ' Down Hand ' buttons at the bottom of the dialog	
7	Click on the OK button to close the dialog; any new event types are available for use in the Project Calendar	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Project Calendar** permission in order to create or delete Project Calendar event subtypes; if security is not enabled, you can change data without this permission

Learn More:

- [Calendar](#)^[392]

3.6.7.2 Allocated Resources

Allocated Resources is a display mode of the project Calendar facility.

In Allocated Resources mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic), with at least one icon representing each element in the project to which a resource has been allocated during that period. If an element has more than one allocated resource, each element:resource combination is represented separately.

Each icon shows the element name and resource name, and indicates the status of the assigned work with one of the following symbols:

- a green square, indicating that the resource has been assigned the work
- a green tick, indicating that the resource has completed the assigned work
- a red square, indicating that the resource has not completed the assigned work
- a square divided diagonally with green and red halves, indicating that the assigned work is to be completed within one working day

Typically, while the work is in progress a resource is represented by:

- An icon with a green square on the day the work is planned to start, and
- An icon with a red square on the day the work is planned to finish

When the work is complete, the icons on both days display the green tick.

Resources are initially allocated to elements through the:

- Resource Allocation tab of the Project Management window, or
- Allocated Work tab of the Personal Information window

You can then *edit* these allocations through either of those windows or through the Allocated Resources Calendar itself.

Access: [View | Project Calendar > Allocated Resources](#)

Use to:

- Review the allocation of resources across the project over a period of time
- Review the status of the work assigned to those resources
- Display and edit the details of the task assigned to a resource
- Allocate further resources to an element
- Delete a resource allocation from an element
- Create and send an email to the resource allocated to an element

How to:

On the Allocated Resources Calendar, make selections from the options as indicated below:

Option	Usage	See also
Display/edit details of an allocated resource	<p>Either:</p> <ul style="list-style-type: none"> • Double-click on any icon for that element:resource combination, or • Right-click on the icon and select the Properties context menu option <p>The Assigned Resources dialog displays, showing the details of the assignment of the resource to a task on the element</p>	<p>Resource Allocation^[349]</p> <p>The Project Management Window^[347]</p>

Option	Usage	See also
	This dialog has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode	
Add resource to element	<p>Right-click on any icon for the element and select the Assign Resource to <element name> context menu option</p> <p>The Assigned Resources dialog displays, showing the start and end dates both set to the date from which you selected the icon, and the Allocated Time field set to 1; all other fields are blank</p> <p>This dialog has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode</p> <p>Specify the resource and the role or task that resource is performing, and define the period for which the resource is allocated to the element</p>	Resource Allocation <small>[349]</small> The Project Management Window <small>[347]</small>
Message Resource	<p>Right-click on any icon for the element:resource combination and select the Message Resource <resource name> context menu option</p> <p>The Model Message dialog displays, on which you create and send your message to the resource allocated to the element</p>	Create a Message <small>[381]</small>
Delete resource from element	<p>Right-click on any icon for the element:resource combination, and select the Delete Resource from <element name> context menu option</p> <p>A prompt displays to confirm the deletion; click on the Yes button</p> <p>The icon and any corresponding icons for that element:resource combination are deleted from the calendar, and the resource is no longer allocated to that element</p>	
Locate element in diagrams in which it is used	<p>Right-click on any icon for the element, and select the Find in all Diagrams context menu option</p> <p>If the element is used in only one diagram, that diagram displays</p> <p>If the element is used in more than one diagram the Element Usage dialog displays, listing the diagrams in which the element occurs</p> <p>Select the required diagram and click on the Open button to display that diagram</p>	Show Element Use <small>[630]</small>
Locate element in Project Browser	<p>Right-click on any icon for the element, and select the Find in Project Browser context menu option</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted</p>	
Locate resource allocation start date corresponding to allocation end date	<p>Firstly, click on the end date icon for the element:resource combination to highlight it; the corresponding start date icon is also highlighted</p> <p>If the start date icon is not in view, right-click on the end date icon and select the Show Start/End context menu option; the</p>	

Option	Usage	See also
	display scrolls to show the start date and the start date cell is highlighted	
Locate resource allocation end date corresponding to allocation start date	<p>Firstly, click on the start date icon for the element:resource combination to highlight it; the corresponding end date icon is also highlighted</p> <p>If the end date icon is not in view, right-click on the start date icon and select the Show Start/End context menu option; the display scrolls to show the end date and the end date cell is highlighted</p>	
Refocus display on today's date	<p>Right-click anywhere on the display and select the Show Today context menu option</p> <p>The display returns to today's date, which is highlighted and outlined</p>	

Notes:

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the above editions, if security is enabled you must have **Manage Project Information** permission in order to allocate resources to or remove resources from the project Calendar; if security is not enabled, you can change data without this permission
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them
- As the start and end icons for a *completed* task are identical, and one-day tasks have only one icon, the methods to show the corresponding start date or end date for an icon are also useful to show whether the icon *is* for a start or end date, or for a one-day task; alternatively, double-click on the icon and review the allocation dates

Learn More:

- [Project Calendar](#)^[390]
- [List of Available Permissions](#)^[205]

3.6.7.3 Project Tasks

Project Tasks is a display mode of the project Calendar facility.

In Project Tasks mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic), with at least one icon representing each *project task* scheduled during that period. The icon represents an independent record of the task - there are no comparisons or validations of the tasks. Therefore you can have separate icons for the same task, with different resources allocated or no resources allocated; you can even have identical task records, if these serve a purpose for you.

Each icon shows the task name and resource name, and indicates the status of the task with one of the following symbols:

- a green square, indicating that the task has been scheduled
- a green tick, indicating that the task is complete
- a red square, indicating that the task is incomplete
- a square divided diagonally with green and red halves, indicating that the task is to be completed within one working day

Typically, while a task is in progress it is represented by:

- An icon with a green square on the day the work is planned to start, and
- An icon with a red square on the day the work is planned to finish

When the work is complete, the icons on both days display the green tick.

Tasks can be created and managed through this view of the Project Calendar, or the:

- Project Tasks tab of the Project Information window, or
- Project Tasks tab of the Personal Information window

Access: [View | Project Calendar > Project Tasks](#)

Use to:

- Review the tasks across the project over a period of time
- Review the progress of those tasks
- Create new tasks
- Display and edit the details of the task
- Allocate a resource to a task
- Delete a task
- Create and send an email to the resource assigned to a task
- Create and send an email to the owner of the task

How to:

On the Project Tasks display, make selections from the options as indicated below:

Option	Usage	See also
Create a task	Either: <ul style="list-style-type: none"> • Double-click on the date cell in which the task is to start • Right-click on the cell and select the Add New Task context menu option, or • Click on the down-arrow next to the first icon in the Calendar toolbar, and select the New Task option The Task Details dialog displays, through which you create the task	Add, Modify and Delete Tasks ^[357]
Display and edit a task	Either: <ul style="list-style-type: none"> • Double-click on the task • Right-click on the cell and select the Properties context menu option, or • Click on the down-arrow next to the first icon in the Calendar toolbar, and select the Properties option The Task Details dialog displays; if necessary, edit the information	Add, Modify and Delete Tasks ^[357]
Allocate a resource to a task	Certain tasks might not initially be assigned to a resource, but might later require specific assignment to finish them off Open the task to edit it, click on the drop-down arrow in the Assigned field and select the required resource	
Delete a task	Right-click on any icon for the required task and select the Delete '<task-name>' context menu option You are prompted to confirm the deletion; click on the Yes button	

Option	Usage	See also
Create and send a message to the task owner	<p>Right-click on any icon for the task and select the Message Owner context menu option</p> <p>The Model Message dialog displays, on which you create and send your message to the user who owns the task (as identified in the Task Details dialog)</p> <p>If the task does not have a defined owner, this option does not operate</p>	Create a Message [381]
Create and send a message to the resource assigned to the task	<p>Right-click on any icon for the task and select the Message Assigned <resource name> context menu option</p> <p>The Model Message dialog displays, on which you create and send your message to the resource assigned to complete the task (as identified in the Task Details dialog)</p> <p>If the task does not have a defined resource, this option does not operate</p>	
Locate task start date corresponding to task end date	<p>Firstly, click on the end date icon for the task to highlight it; the corresponding start date icon is also highlighted</p> <p>If the start date icon is not in view, right-click on the end date icon and select the Show Start/End context menu option; the display scrolls to show the start date and the start date cell is highlighted</p>	
Locate task end date corresponding to task start date	<p>Firstly, click on the start date icon for the task to highlight it; the corresponding end date icon is also highlighted</p> <p>If the end date icon is not in view, right-click on the start date icon and select the Show Start/End context menu option; the display scrolls to show the end date and the end date cell is highlighted</p>	
Refocus display on today's date	<p>Right-click anywhere on the display and select the Show Today context menu option</p> <p>The display returns to today's date, which is highlighted and outlined</p>	

Notes:

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them

Learn More:

- [Project Calendar](#) [390]
- [Project Tasks](#) [350]
- [Monitor Your Tasks](#) [385]

3.6.8 Use Case Estimation

Project estimation is the task of working out how much time and effort is required to build and deploy a solution.

The Use Case metrics facility in Enterprise Architect provides a starting point for estimating project effort;

using this facility you can get a rough measure of the complexity of a system and some indication of the effort required to implement the model. Like all estimation techniques, Use Case metrics requires some experience with previous projects to 'calibrate' the process.

There is additional information available on Use Case metrics on the Sparx Systems website.

Topics:

Topic	Detail	See also
Calibrating	<p>The following values must be carefully calibrated in order to gain the best possible estimates:</p> <ul style="list-style-type: none"> • Technical Complexity Factors, which are values that attempt to quantify the difficulty and complexity of the work in hand • Environment Complexity Factors, which are values that attempt to quantify non-technical complexities such as team experience and knowledge • Default Hour Rate, which sets the number of hours per Use Case point 	<p>Technical Complexity Factors ^[401]</p> <p>Environment Complexity Factors ^[402]</p> <p>Default Hours ^[404]</p>
Estimating	<p>Once you have entered all the calibration values, you can estimate the project timescale through the Use Case Metrics dialog</p>	<p>Estimating Project Size ^[404]</p>

Learn More:

- [Project Estimation Using Use Case Metrics](#)

3.6.8.1 Technical Complexity Factors

Technical Complexity Factors (TCFs) are used in the Use Case Metrics estimation technique.

The EABase.eap model contains a default set of TCFs, which you can add to or modify using the Estimation Factors dialog. This set of factors should include all factors that could affect the technical complexity of the project environment.

Access: [Settings](#) | [Project Types](#) | [Estimation Factors](#) > [Technical Complexity Factors](#)

How to:

To maintain Technical Complexity Factors, follow the steps below:

Step	Action	See also
1	<p>On the Technical Complexity Factors tab, either:</p> <ul style="list-style-type: none"> • Click on the New button to add another TCF, or • Click on the required factor in the Defined Technical Types list to edit it (go to step 3) 	
2	<p>In the Factor Number field, type the appropriate TCF number</p>	

Step	Action	See also
3	In the Description field, type or edit the TCF description	
4	In the Weight field, type or edit the technical complexity weighting This indicates how much technical complexity you assign to a factor; for example, <i>the system is to be developed in ADA</i> might warrant a higher weight than <i>the system is to be a shell script</i>	
5	In the Value field, type or edit a value representing the degree of influence the factor has on the project As a suggested gauge: <ul style="list-style-type: none"> • 0 indicates no influence • 3 indicates average influence • 5 indicates strong influence 	
6	Click on the Save button	
7	Examine the Defined Technical Types list, and scroll across it to show the Ex Value column (Weight x Value) The summed Ex Values yield the Unadjusted TCF value (at the bottom of the dialog) The Unadjusted TCF value is combined with the Environment Complexity Factors to skew the overall complexity up or down, depending on the level of technical complexity and the corresponding level of environmental support	Environment Complexity Factors ^[402]

Notes:

- The TCF Weight evaluates its respective factor, but is irrelevant to a project; the **Value** field assesses each factor's role within a project and, for most purposes, is the only field requiring adjustment
- The supplied factors and their associated weights are defined by the *Use Case Points Method*, although they can be adjusted to suit a project's specific requirements
- You can transport the Technical Complexity Factors between models, using Export Reference Data and Import Reference Data

Learn More:

- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

3.6.8.2 Environment Complexity Factors

Environment Complexity Factors (ECFs) are used in the Use Case Metrics estimation technique.

The EABase.eap model contains a default set of ECFs, which you can add to or modify using the Estimation Factors dialog. This set of factors should include all factors that could affect the general design and development environment, including team experience and knowledge, team size, expertise and other non-functional environmental factors.

Access: [Settings | Project Types | Estimation Factors > Environment Complexity Factors](#)

How to:

To maintain Environment Complexity Factors, follow the steps below:

Step	Action	See also
1	On the Environment Complexity Factors tab, either: <ul style="list-style-type: none"> Click on the New button to add another ECF, or Click on the required factor in the Defined Environment Types list to edit it (go to step 3) 	
2	In the Factor Number field, type the appropriate ECF number	
3	In the Description field, type or edit the ECF description	
4	In the Weight field, type or edit the environment complexity weighting This indicates how much complexity you assign to a factor	
5	In the Value field, type or edit a value representing the degree of influence the factor has on the project As a suggested gauge: <ul style="list-style-type: none"> 0 indicates no influence 3 indicates average influence 5 indicates strong influence 	
6	Click on the Save button	
7	Examine the Defined Environment Types list, and scroll across it to show the Ex Value column (Weight x Value) The summed Ex Values yield the Unadjusted ECF value (at the bottom of the dialog) The Unadjusted ECF value is combined with the Technical Complexity Factors to skew the overall complexity up or down, depending on the level of technical complexity and the corresponding level of environmental support	Technical Complexity Factors ^[40]

Notes:

- The ECF Weight evaluates its respective factor, but is irrelevant to a project; the **Value** field assesses each factor's role within a project and, for most purposes, is the only field requiring adjustment
- The supplied factors and their associated weights are defined by the *Use Case Points Method*, although they can be adjusted to suit a project's specific requirements
- You can transport the Environment Complexity Factors between models, using Export Reference Data and Import Reference Data

Learn More:

- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

3.6.8.3 Default Hours

Setting an hourly rate is the most difficult factor in an accurate estimation. Typical ranges can vary from 10 to 30 hours per Use Case point.

Studying the *Use Case Points Method*, from which this variable is defined, can help you to understand its role in the estimation and facilitate selection of a suitable initial value.

The best way to estimate this value is through analysis of previous completed projects. By calculating the project estimation on a completed project for which the Use Cases and environment are configured within Enterprise Architect, you can adjust the hour rate to render an appropriate value for your unique work environment.

Access: [Settings | Project Types | Estimation Factors > Default Hour Rate](#)

How to:

To set the default hour rate per adjusted Use Case point, follow the steps below:

Step	Action	See also
1	In the Duration field, type the number of hours per Use Case Point	
2	In the Hourly Rate field, type the cost per hour	
3	Click on the Close button	

Notes:

- The values you enter are stored as local settings on your computer only
- This option is also active in the 'Lite', read-only version of Enterprise Architect

Learn More:

- [Project Estimation Using Use Case Metrics](#)
- [The Read-only 'Lite' Edition](#) ^[18]

3.6.8.4 Estimating Project Size

Enterprise Architect uses a simple estimation technique based on the established:

- Number of Use Cases to be built
- Difficulty level of those Use Cases
- Project environment factors and
- Build parameters

This technique is of value only once you have developed a couple of known projects to use as a baseline. Please DO NOT use the provided 'guesstimates' as a real world measure until you have some real world base lines to measure against.

Access: Select package in Project Browser:

Project | Use Case Metrics
Right-click | Documentation | Package Metrics

How to:

On the Use Case Metrics dialog, work through the fields as indicated in the following table:

Field	Usage	See also
Root Package	Confirm the root package in the hierarchy All Use Cases under this package could potentially be included in the report	
Reload	Re-run the load from the selected package, usually after you change the filter criteria	
Phase like	Include Use Cases with a phase that matches the wildcard value in the field Use * to match any characters, for example 1.* for 1.1 and 1.2	
Keyword like	Include Use Cases with a keyword that matches the wildcard value in the field Use * to match any characters	
Bookmarked	Include all Use Cases, or only those that are tagged, or those that are not tagged	
Use Cases	Check the total count of Use Cases in the estimate The Use Cases and their parameters are listed in the panel underneath this field	
Include Actors	Select to include Actors in the estimate	
Technical Complexity Factor	Review the parameters that describe the degree of technical complexity of the project While the Unadjusted TCF Value comes from the Technical Complexity Factors tab of the Estimation Factors dialog, the other values compose the Use Case Points Method formula Modify these fields with caution The final project estimate is directly proportional to the TCF	Technical Complexity Factors [401]
Environment Complexity Factor	Review the parameters that calculate the degree of environmental complexity of the project, from factors such as programmer motivation or experience The listed parameters compose the formula calculating the ECF, defined by the Use Case Points Method; the only parameter affected by the project is the Unadjusted ECF Value , derived from the Environment Complexity Factors [317] tab of the Estimation Factors dialog The final project estimate is directly proportional to the ECF	Environment Complexity Factors [402]
Unadjusted Use Case Points (UUCP)	Check the sum of the Use Case complexity numbers	
Ave Hours per Use Case	Check the average number of hours assigned to easy, medium and difficult Use Cases You cannot change these figures	

Field	Usage	See also
Total Estimate	Review the detailed breakdown of the final figure You must tailor the hours per Use Case Point figure to the level that matches your type of project and capability based on known previous project outcomes	
Default Rate	Set the default hours ^[404] to be fed into the final calculation	Default hours ^[404]
Re-Calculate	Re-run the estimate, usually after you change the hours or Use Case point number	
Report	Produce a rich text formatted report from the current estimate	

3.6.9 SPEM

Topics:

Topic	Detail	See also
Introduction	<p>According to the Object Management Group (OMG) Software & Systems Process Engineering Meta-Model Specification (Version 2.0, April 01 2008):</p> <ul style="list-style-type: none"> <i>The Software and Systems Process Engineering Meta-model (SPEM) is a process engineering meta-model as well as conceptual framework, which can provide the necessary concepts for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes. An implementation of this meta-model would be targeted at process engineers, project leads, project and program managers who are responsible for maintaining and implementing processes for their development organizations or individual projects.</i> <p>In 1999, the OMG placed a Request for Proposal concerning Software Process Engineering (SPE). In November 2002, the OMG released the Software Process Engineering Meta-model Specification 1.0. SPEM was defined as a Profile of UML, which used UML as a notation and took an object-oriented approach. To accommodate UML 2, the SPEM specification was upgraded to 2.0 in April 2008.</p>	http://www.omg.org/spec/SPEM/2.0/PDF (Online Resource)
SPEM in Enterprise Architect	<p>SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM, to describe an actual development process.</p> <p>Enterprise Architect enables you to develop SPEM diagrams quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The SPEM facilities are provided in the form of:</p> <ul style="list-style-type: none"> A SPEM diagram type, accessed through the New Diagram dialog A set of SPEM pages in the Toolbox, providing SPEM elements (stereotyped UML elements) SPEM element and relationship entries in the 	Add New Diagram ^[568] SPEM Toolbox ^[407] Toolbox Shortcut Menu ^[551] Quick Linker ^[622]

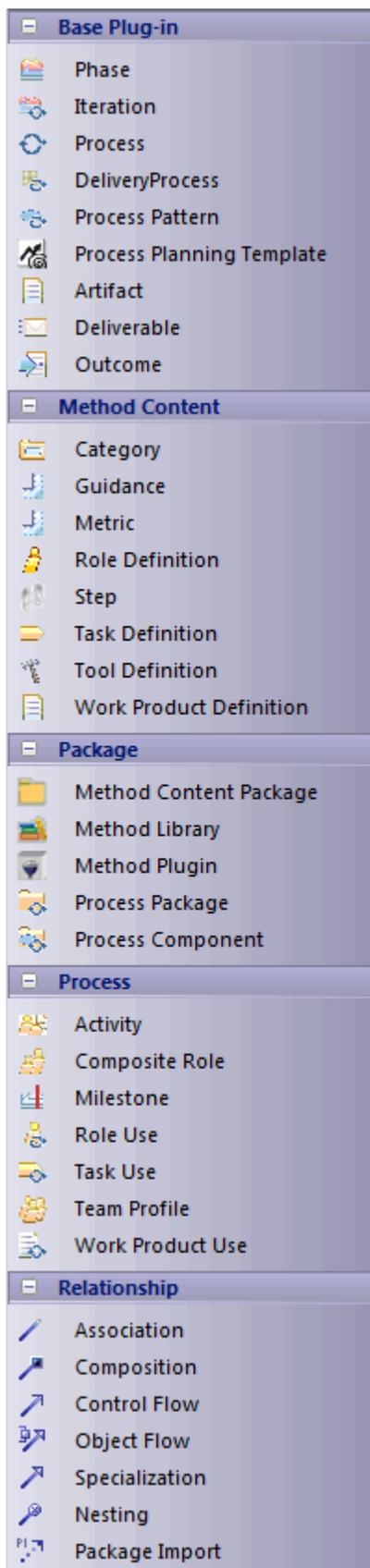
Topic	Detail	See also
	Toolbox Shortcut Menu and Quick Linker	
Disable SPEM	If you prefer not to use SPEM in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	MDG Technologies 

3.6.9.1 SPEM Toolbox Pages

These pages provide the graphical SPEM elements for drawing the diagrams.

Access: From the Diagram Toolbox: More Tools | SPEM

Example:



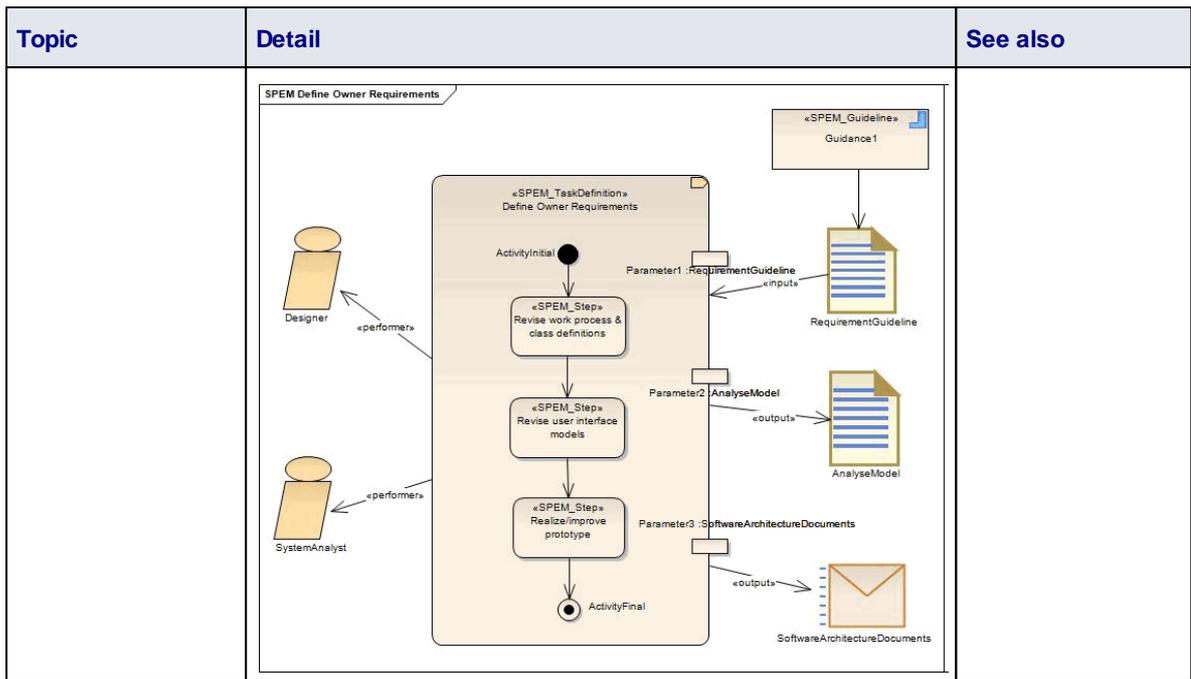
Reference:

Page	Item	Action
Base Plug-in	Phase	Create a predefined special Activity representing a significant period in a project.
	Iteration	Group a set of nested Activities that are repeated more than once. Typically, Iteration is an Activity for which the default value of the <i>isRepeatable</i> attribute is True .
	Process	Represent a special Activity that describes a structure for particular types of development projects, or parts of them.
	Delivery Process	Represent a special Process describing a complete and integrated approach for implementing a specific project type.
	Process Pattern	Represent a special Process to describe a reusable cluster of Activities in a general process area that provides a consistent development approach to common problems.
	Process Planning Template	Represent a special Process that is prepared for instantiation by a project planning tool.
	Artifact	Represent a Work Product Definition that provides a description and definition for tangible work product types.
	Deliverable	Represent a Work Product Definition that provides a description and definition for packaging other Work Products, and that can be delivered to an internal or external party.
	Outcome	Represent a Work Product Definition that provides a description and definition for non-tangible work products.
Method Content	Category	Categorize content based on the user's criteria.
	Guidance	Identify reference items such as Guidelines, Templates, Checklists, Tool Mentors, Estimates, Supporting Materials, Reports and Concepts.
	Metric	Define a standard measurement for instances of Method Content elements.
	Role Definition	Define a set of related skills, competencies, and responsibilities.
	Step	Represent parts or subunits of a Task Definition.
	Task Definition	Describe an assignable unit of work. Every Task Definition is assigned to specific Role Definitions. A Task is associated with input and output Work Products.
	Tool Definition	Describe the tools that are recommended or necessary for completing a specific Task.
	Work Product Definition	Define any forms of document, report or outcome that are consumed, produced or modified by Tasks.
Package	Method Content Package	Create a physical container to organize the Method Content elements.

Page	Item	Action
	Method Library	Create an overall physical container for all SPEM 2.0 elements.
	Method Plugin	Create a physical container for Method Content Package and Process Packages. It can be used stand-alone as well as extended to many other Method Plugins.
	Process Package	Create a physical container that contains different kinds of Process element.
	Process Component	Create a special Process Package that provides the mechanism of encapsulation.
Process	Activity	Define basic units of work within a Process as well as the Process itself.
	Composite Role	Represent an aggregation of Role Definition references for an Activity.
	Milestone	Represent any significant events in a development project.
	Process	Create a special Activity that describes a structure for particular types of development project.
	Role Use	Represent a Role Definition in the context of one specific Activity.
	Task Use	Represent a Task Definition in the context of one specific Activity.
	Team Profile	Define a nested hierarchy of teams and team members.
	Work Product Use	Represent a Work Product Definition in the context of one specific Activity.

Topics:

Topic	Detail	See also
Presentation	<p>In Enterprise Architect, every SPEM stereotype can be presented in one of two ways:</p> <ul style="list-style-type: none"> • Iconic presentation, or • Textual presentation. <p>The <i>iconstyle</i> tag is used for switching between these presentations. For example, in the SPEM diagram below, if you want the <i>SPEM_TaskDefinition</i> to have iconic presentation you set the <i>iconstyle</i> Tagged Value to True, and display the element as an icon. To get the textual presentation for <i>SPEM_TaskDefinition</i>, as an outline with a decoration in the top right corner, set the Tagged Value to False.</p>	



3.6.10 Update Package Status

Elements in Enterprise Architect can be assigned a current status, such as *Proposed*, *Validated* or *Mandatory*.

Often a complete package structure moves from one status to another (such as for release) in one operation. To help facilitate this, Enterprise Architect supports a 'bulk' update of element status.

Access: [Project Browser package context menu | Package Control | Update Package Status](#)

Use to:

- Update the status of all elements in a package at the same time

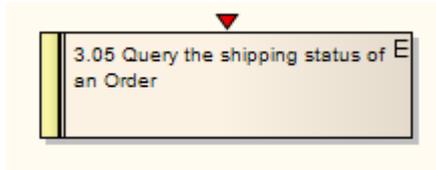
How to:

To update the status of all elements in a package, using the Status Update dialog, follow the steps below:

Step	Action	See also
1	Select: <ul style="list-style-type: none"> • The new status • Whether to recursively descend the package tree • Whether to include elements • Whether to include element requirements • Whether to include element constraints 	
2	Click on the OK button Enterprise Architect updates all required elements to the new status	

3.6.11 Manage Bookmarks

A bookmark is a visual clue that something is different about an element; it is represented by a small red triangle that displays above the element in the diagram.



You can assign any meaning to the bookmark as is appropriate to your model.

You can bookmark:

- Single elements manually, or
- All elements in a package automatically when they assume a defined status

How to:

To work with bookmarks, select from the options described below:

Option	Usage	See also
Bookmark a selected element in a diagram	Either: <ul style="list-style-type: none"> • Press (Shift+Spacebar), or • Select the Edit Bookmark Selected menu option 	
Clear a bookmark on a selected element	Use either of the above two options again	
Bookmark Multiple Elements in a Package	Select Project Browser package context menu Bookmarks The Manage Bookmarks dialog displays, enabling you to automatically bookmark any elements (and, if required, their children) in the package that have: <ul style="list-style-type: none"> • New changes defined in the Maintenance window • New defects defined in the Maintenance window, and/or • Test scripts defined in the Testing window This is useful to highlight elements that have additional project information	
Clear all elements of bookmarks	To clear all bookmarks: <ul style="list-style-type: none"> • Throughout the current package, click on the Clear All button on the Manage Bookmarks dialog • In the current diagram, select the Edit Clear All Bookmarks menu option 	

Notes:

- You should reload the project to update it with the new or cleared bookmarks
- The Model Search window enables searching based on bookmarked elements

Learn More:

- [The Maintenance Workspace](#)^[1713]
- [The Testing Workspace](#)^[1693]
- [Refresh View of Shared Project](#)^[187]
- [Create & Modify Searches](#)^[482]

3.6.12 Monitor Change Events

You can automatically monitor work events in your project, using the Model Views facility.

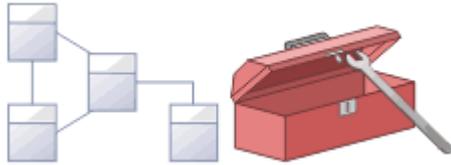
Guide:

Topic	Detail	See also
Model Views	<p>This facility enables you to:</p> <ul style="list-style-type: none"> • Automatically refresh a search in a View at an interval that you define • Notify you if the results of the search change between two consecutive searches <p>You can therefore use Model Views to monitor various events in the development project, depending on how you set up the search in a View</p>	Model Views Operations ^[469]
Example	<p>You could set up a search to detect:</p> <ul style="list-style-type: none"> • Change items, or Issue items, so that Enterprise Architect would notify you as new items were created • Element Status, Type, Phase, Version, Priority and/or date of last update, so that Enterprise Architect would notify you as items were progressed to: <ul style="list-style-type: none"> • Fall in to the level of work represented by the search categories or • Move out of the categories into the next level of work • Tagged Values, so that - again - as items were changed to satisfy the criteria of a sequence of searches, the progression of items through a set of stages could be checked and managed 	
Personalization	<p>People responsible for different stages in a process could have their own Model View searches so that:</p> <ul style="list-style-type: none"> • As a development, validation or authorization task falls due the responsible person is automatically notified, and • When the work is complete both the next person in line and the overseeing manager are notified 	

Notes:

- This facility is available in the Enterprise Architect Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

3.7 Project Maintenance



This topic highlights some administrative functions you might have to carry out to maintain your model.

Topics:

Topic	Detail	See also
Check the integrity of the data in a project	If you have a failed XML import, network crash or other unforeseen event that could disrupt the integrity of information in the model, it is recommended that you run a Project Integrity Check	Check Project Data Integrity ^[414]
Reset Auto Increment Columns in Tables	XML Import/Export can affect the table auto increment column values and push them towards the maximum datatype value; you can re-sequence the columns to avoid this problem	Reset Table Auto Increment or Identity Columns ^[416]
Upgrade an old project to enable use of new features	The structure of Enterprise Architect project files is occasionally changed to support more features; when this happens, existing project files must be upgraded to the new format to ensure correct operation and to take advantage of all the new features	Upgrade Projects ^[417]
Rename a project	If you want to rename an Enterprise Architect project, you can do so through Windows Explorer	Rename a Project ^[418]
Compact a project	After some time, a project .EAP file might benefit from compacting to conserve space	Compact a Project ^[418]
Repair a project if it did not close properly	If a project has not been closed properly, such as during system or network outages, on rare occasions the .EAP file does not re-open correctly.	Repair a Project ^[419]

Notes:

- You only rename, compact and repair models created as .EAP files; these processes are not required for models stored in a DBMS

3.7.1 Check Project Data Integrity

If you have a failed XML import, network crash or other unforeseen event that could disrupt the integrity of information in the model, it is recommended that you run the *Project Integrity Check* function.

You can select a variety of items to check. The integrity check examines all database records and ensures there are no 'orphaned' records or inaccurate or unset identifiers. This function does NOT check UML conformance, only the data relationships and repository structure.

You can run the integrity checker first in report mode to discover if anything should be corrected, and then run it again in 'recover/clean' mode.

When Enterprise Architect 'cleans' the model, it attempts to recover lost packages and elements, and generates a new package at the model root level called `_recovered_`. Check through any elements that are found and, if required, drag them into the model proper. If they are not required, delete them.

Access: [Tools](#) | [Data Management](#) | [Project Integrity Check](#)

Use to:

- Ensure that your project data is structurally complete

How to:

To check the data integrity of your project, follow the steps below:

Step	Action	See also
1	Select the Project Integrity Check menu option The Project Integrity Check dialog displays	
2	Select the checkbox for each check to run: <ul style="list-style-type: none"> • Package Structure • Object Structure • Object Features • All GUIDs • Cross References • Connectors • UML 2.0 Migration (see below) 	
3	Select either: <ul style="list-style-type: none"> • the Report Only option to just view a report on the state of your model, or • the Recover/Clean option to recover and clean your project; before selecting this option back up your project file first 	
4	To write a log of the integrity check, click on the Save Results button and select a log file	
5	Click on the Go button to run the check If you want to display the resulting information in a more readable layout, you can resize the dialog and its columns	

UML 2.0 Migration

The UML 2.0 Migration check enables you to migrate the project from UML 1.3 semantics to UML 2.0 semantics. The migration process currently converts activities that are invocations of operations into called operation actions as per the UML 2.0 specification.

The UML 2.0 Migration option is an exclusive process that does not enable any of the other checks to be selected. When you click on the **Go** button to perform the migration, a prompt displays for you to confirm the operation.

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Check Data Integrity** permission to perform a data integrity check

Learn More:

- [List of Permissions](#)^[205]
- [Run SQL Patches](#)^[418]

3.7.2 Reset Table Auto Increment or Identity Columns

This topic explains possible impacts of XML Export/Import on table auto increment columns, and how to re-sequence the columns whose value approaches the maximum datatype value.

Topics:

Topic	Detail	See also
XML Export/Import	<p>XML Export/Import can cause gaps in the numbering sequence of auto increment columns</p> <p>Each XML Import deletes rows from several tables; the import then adds rows starting from the maximum value of the auto increment column</p> <p>Repeated XML imports can result in the value of the auto increment approaching the maximum value of the database datatype; for example, SQL Server's <i>int</i> datatype has a maximum value of 2,147,483,647</p>	
Replication	<p>Large auto increment values can also arise where the project originated as an EAP replica or design master</p> <p>The Jet engine assigns random values for auto increment columns with each XML Import into the project</p> <p>These random values can approach the maximum range of the repository data type, which could present a problem when the EAP project is transferred to a repository</p>	Replication ^[188]

Access: [Tools](#) | [Data Management](#) | [Reset IDs](#)

How to:

To re-sequence auto increment columns, follow the steps below:

Step	Action
1	Open the project
2	<p>Select the Reset IDs menu option</p> <p>A dialog displays listing all non-empty tables that contain an auto increment or identity column</p> <ul style="list-style-type: none"> • The Rows column shows the number of rows in the table • The Maximum ID column shows the current maximum value of the auto increment column • The Action column shows either No Action or Reset, depending on how close the column value is to the datatype's maximum

Step	Action
	Tables requiring a reset are automatically selected in the list
3	Click on the Go button to reset the auto increment column values

3.7.3 Project Upgrade

The structure of Enterprise Architect project files is occasionally changed to support more features. In such cases, the project might have to be upgraded.

Topic	Detail	See also
Upgrading	Upgrading to the new file structure is a simple and quick process that brings your project to the current level to: <ul style="list-style-type: none"> • Ensure correct operation and • Support all the latest Enterprise Architect features 	
Initial Check	When you load a project that was created in an early release of Enterprise Architect (for example, an archived project) using a recent release of Enterprise Architect, the system determines whether the project should be upgraded and, if the upgrade is necessary, displays the Upgrade Wizard	
The Upgrade Project Wizard	The Upgrade Project Wizard takes you through the upgrade process and: <ul style="list-style-type: none"> • Advises you of the necessity to upgrade • Advises you to back up the current project; it is essential to back up before any changes are made • Checks which upgrade path is required • Guides you through the steps to perform the upgrade • Opens the newly converted project 	

Notes:

- If you are using replication in your models, and the upgrade wizard detects that the project you are opening is a replica and not a Design Master, a different upgrade path is required
- Once upgraded, the project cannot be opened with the version of Enterprise Architect in which it was created

Learn More:

- [Replication](#)^[188]
- [Upgrade Replicas](#)^[417]

3.7.3.1 Upgrade Replicas

Models that have replication features added might have to be upgraded differently from regular projects.

- If the model is a Design Master (the root model of all other replicas) then you can upgrade the model to suit the current version of Enterprise Architect; after upgrading a Design Master you should re-create the replicas, rather than synchronizing
- If the model is not a Design Master, Enterprise Architect must first remove the replication features, then upgrade the project in the normal manner; the Upgrade Wizard guides you through the steps

Learn More:

- [Project Upgrade](#) ^[417]
- [Replication](#) ^[188]
- [Design Masters](#) ^[190]

3.7.4 Run SQL Patches

Occasionally, Sparx Systems might release a patch to correct a model fault.

Access: [Tools](#) | [Data Management](#) | [Run Patch](#)

The patch generally checks how many records are to be updated, and reports on what is to be done.

3.7.5 Rename a Project

If you want to rename a project, you can only do so at the Windows file system level, using Windows Explorer.

How to:

To rename an Enterprise Architect project .EAP file, follow the steps below:

Step	Action
1	If you have the project open, shut it down
2	Ensure no other users have the file open
3	Open Windows Explorer and navigate to the project
4	Rename the project file using Windows Explorer
5	You should keep the .EAP extension the same to preserve compatibility with the default project type, as installed in the registry at installation time

3.7.6 Compact a Project

After a long period of use and development, a project .EAP file might benefit from compacting to conserve space.

Always compact and repair projects on a local drive, never on a network drive.

Access: [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Compact .EAP File](#)

How to:

To compact a project, follow the steps below:

Step	Action
1	If you have the project open, shut it down
2	Ensure no other users have the file open
3	Select the Compact .EAP File menu option

Step	Action
4	Follow the on-screen instructions to complete the process

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Administer Database** permission to compact a project

Learn More:

- [List of Permissions](#)^[205]

3.7.7 Repair a Project

If a project has not been closed properly, such as during system or network outages or on poor network connections, on rare occasions the .EAP file does not open correctly. A message displays informing you the project is of an unknown database format or is not a database file.

You can repair the project; however, never attempt to repair a project over a network connection; copy it to a local drive first.

Access: [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Repair .EAP File](#)

How to:

To repair a project that was not closed properly, follow the steps below:

Step	Action	See also
1	Ensure that all users are logged off the project you are attempting to repair	General ^[422]
2	Copy the project file to a local drive on your PC	
3	In Enterprise Architect, select Tools Options to display the Options dialog On the General page, deselect the Use Jet 4.0 - requires restart checkbox	
4	Close and restart Enterprise Architect and open a <i>copy</i> of the project you intend to repair as a <i>place holder</i> project, to enable access to the Repair .EAP File facility	
5	Select the Repair .EAP File menu option, and follow the on-screen instructions	
6	Once you have repaired the project, it is recommended that you perform a data integrity check	Check Project Data Integrity ^[414]

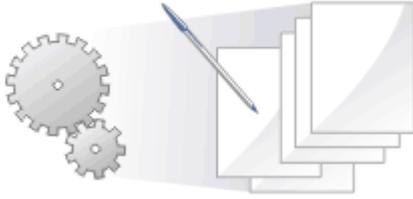
Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Administer Database** permission to repair a project

Learn More:

- [List of Permissions](#)^[205]

3.8 Local Options



There are many options for customizing how Enterprise Architect displays and works with models and model elements. This topic describes those settings that are local to a particular user and machine.

Most of these settings are stored in your registry so they are set for your use only. For a networked workplace, registry settings can be copied down to any network workstation you log in to. Otherwise, the settings are valid for the current machine only.

Access: [Tools | Options](#)

Select the required page of options by clicking on the appropriate category name in the left hand list on the dialog. For information on the options on a specific page, select the appropriate page title link below.

Topic	Detail	See also
General Options	Set options that define how Enterprise Architect operates as a whole	General ^[422]
Standard colors of objects	Define the display and background colors of objects and properties across Enterprise Architect	Standard Colors ^[424]
Diagram actions	Configure overall options for new diagrams and general diagram behavior	Diagram ^[426]
Diagram appearance	Define how diagrams and their contents appear on display	Diagram Appearance ^[427]
Diagram behavior	Define how a diagram responds to actions taken on it	Diagram Behavior ^[430]
Options for Sequence diagrams	Configure various font settings and the focus of the control indicator for Sequence diagrams	Diagram Sequence ^[431]
Object appearance	Configure how elements look and respond in diagrams	Objects ^[432]
Link or Connector appearance	Define options for the creation, behavior and notation of connectors	Links ^[434]
Colors of communication messages	Configure the colors used in Communication diagrams	Communication Message Colors ^[435]
XML Specification options	Configure various settings for working with XML	XML Specifications ^[436]
Source code engineering options	Set general options that apply to all languages when generating code from Enterprise Architect, including the default language in which to generate code	Source Code Engineering ^[1517]
DDL and code	Configure options for Enterprise Architect's internal editor,	Code Editors ^[1519]

Topic	Detail	See also
editors	and for the default editor for DDL scripts	
Constructor/ Destructor options	Configure options for Object Lifetimes	Object Lifetimes ^[1521]
Attribute and Operation specifications	Configure Attribute and Operation specifications	Attribute/Operations ^[1522]
Code language specifications	Define how Enterprise Architect handles a particular language when generating code, including the default source directory and the default editor for a specific code language	ActionScript ^[1530]
		C ^[1531]
		C# ^[1532]
		C++ ^[1533]
		Delphi ^[1533]
		Java ^[1535]
		PHP ^[1536]
		Python ^[1536]
		Visual Basic ^[1539]
		VB.Net ^[1537]

3.8.1 General

Access: **Tools | Options > General**

Use to:

- Set general options

Reference:

Field	Usage	See also
Author	Set the default author when new elements are created and modifications made	
Clipboard Format	Set the graphic format in which to save image to the clipboard; Metafile has the best detail	
Double-click on Browser	Configure the Project Browser behavior	Set Default (Project Browser) behavior ^[455]
Use Jet 4.0 - requires restart	Set JET 4.0 as the database engine; this ensures compatibility with .EAP files that are in turn compatible with versions of MS Access later than Access 97, and that support unicode character sets If your project is not in a Jet 4.0 database, you should also download a copy of the Jet 4.0 EABase Model from the Sparx	Perform a Project Data Transfer ^[343] Download link: Sparx Systems website

Field	Usage	See also
	Systems website, and do an EAP to EAP transfer of your model into the Jet 4.0 file	
Use extended « and » characters	Apply the guillemet characters to stereotypes For some double byte character sets, it is best to select this checkbox.	
Allow custom RTF tags in reports	Enable you to use customized rich text format code in report templates when generating reports with the Legacy RTF Report Generator From release 7.0 of Enterprise Architect, with the Notes formatting facility, this option is not really necessary	Legacy RTF Report generator ^[176] Notes toolbar ^[770]
Project Directory	Specify the default location of Enterprise Architect projects	
File Directory	Specify the default location for file	
Web Home	Specify the default home page to open when you click on the Home button in the internal web browser	
Web Search	Specify the default web page to open when clicking on the Web Search button in the internal web browser	The Web Browser ^[128]
Web Mail	Specify the email server address (http://xxxx/exchange/) for accessing email through the web browser within Enterprise Architect	
Confirm Deletes	Use or bypass the Confirm Delete dialog; only clear this checkbox if you are an experienced user	
Allow Free Sorting	Enable you to re-order elements within a package regardless of element type, in the Project Browser	
Show Stereotypes	Show element and feature stereotypes in the Project Browser	
Always open maximized	Ensure that Enterprise Architect always starts up in a maximized window	
Hide Properties Info Section	Hide or show the properties information status bar on the Properties window	
Docked Windows Tabs on Bottom (requires restart)	Display the docked window tabs at the bottom of the window (default) Clear the checkbox to show the tabs at the top of the windows	
Main Diagram Tabs at Bottom (requires restart)	Display the diagram tabs at the bottom of the main view (default) Clear the checkbox to show the tabs at the top of the main view	
File Loading Strategy	Select the Enterprise Architect model loading behavior for .EAP files; choose either: <ul style="list-style-type: none"> • Load on Demand (Lazy Load) • Preload Entire Model Load on Demand does not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded	

Field	Usage	See also
	<p>This means that a model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model</p> <p>For server connections, the Load on Demand option is set when the connection is created</p>	

3.8.2 Standard Colors

The Standard Colors page of the Options dialog enables you to set the display color of a range of objects and their backgrounds. On first use, the page displays the system default colors. To display the range of colors available for an item, or to define a new color, click on the down arrow at the end of the appropriate field.

Access: [Tools | Options > Standard Colors](#)

Use to:

- Set global colors

Reference:

Field	Usage	See also
Paper	Define the paper (background) color in diagrams	
Element Fill	Define the fill color of elements	
Element Line	Define the line color of elements	
Shadow	Define the color of element outline shadows	
Attribute	Define the color of attribute text	
Method	Define the color of method (operation) text	
Note Color	Define the note background color	
Note Compartment Text	Define the color of text in the element Note compartment	
Screen	Define the screen (element) color	
Connector Line	Define the connector line color	
Behavior	Define the color for behaviors in Activity diagrams	
Custom Colors	<p>Click on the Define Colors button to define up to 16 custom colors to be used in changing the appearance of elements on diagrams</p> <p>The Show Project Custom Colors in Color Combo Boxes checkbox must be selected in order for this button to operate</p>	Create Project Colors ^[425]
Show Project Custom Colors in Color Combo Boxes	<p>Enable the use of project custom colors within the Default Appearance dialog</p> <p>When you select this option, the last 16 squares on the color selection palettes on the Default Appearance dialog are reserved</p>	Set an Element's Default Appearance ^[641]

Field	Usage	See also
	for the custom colors You must restart Enterprise Architect for changes to this option to take effect	

Notes:

- Using this page of the Options dialog, you can set the background of all diagrams to be a specific color; you can also use the Diagram Appearance page to set all diagram backgrounds to be either a uniform color or to have a fade gradient from top to bottom
- As an alternative, you can create a background image for the diagram
- To override the default appearance of a specific element on all diagrams on which it is found, right-click on the element and select the **Appearance | Default Appearance** context menu option; the Default Appearance dialog displays
- To change the appearance of a specific element on the current diagram only, use the Format toolbar; if this toolbar is not displayed, select the **View | Toolbars | Format Tool** menu option

Learn More:

- [Diagram Appearance](#)^[42]
- [Create Custom Diagram Background](#)^[595]
- [Default Appearance](#)^[64]
- [Format Toolbar](#)^[54]

3.8.2.1 Create Project Colors

This topic explains how you set the project colors to be made available in the element Default Appearance dialog.

Access: **Tools | Options > Standard Colors: Define Colors**

Use to:

- Set the project colors

How to:

To define the custom colors for a project, follow the steps below:

Step	Action	See also
1	Click on the Define Colors button The Color dialog displays	
2	Click on the Define Custom Colors » button The Color dialog is extended to provide color selection fields on the right of the dialog	
3	Click on an empty block in the Custom Colors palette (or, if you are <i>replacing</i> a color, click on that color block in the palette)	
4	Create the required project color using the Luminosity slide bar, color mixer panel, Hue/Sat/Lum fields and/or Red/Green/Blue fields	

Step	Action	See also
5	Click on the Add to Custom Colors button to add the color to the selected Custom colors block	
6	Repeat steps 3, 4, and 5 for any other project colors you want to define	
7	Click on the OK button to close the Color dialog, then click on the Close button to close the Options dialog The custom colors are made available for use in the last 16 squares on the color selection palettes on the Default Appearance dialog	

3.8.3 Diagram

The **New Diagram Defaults** page of the **Options** dialog enables you to configure overall options for new diagrams and general diagram behavior.

Access: **Tools | Options > Diagram**

Use to:

- Configure diagram options and behaviours

Reference:

Field	Usage	See also
Default Page Size	Show the default page size for new diagrams, which you set by clicking on the Page Setup button to display the Page Setup dialog.	
Print with Border	Print pages with a border.	
Landscape	Print pages in landscape orientation. This checkbox is controlled from the Page Setup dialog.	
Show Public Features Show Protected Features Show Private Features	Set the default visibility of Class features.	
Show Diagram Notes	Display the diagram details in the top left corner of all diagrams in the model; details include diagram name, package, version and author.	
Use Opaque Diagram Labels	Specify where opaque diagram labels should display. Screen and Printing are best, Clipboard and Files might not be desirable.	
Strict UML Syntax	Enforce compliance with UML syntax when adding new connectors and other structures The syntax check is not performed for stereotyped elements that have had the <code>_metatype</code> attribute set	Define a Stereotype as a Metatype ^[1054]
Disable fully scoped object names	Disable fully scoped object names, when an element is in a diagram; don't use when the element is in its home package.	

Field	Usage	See also
	A scoped name is of the format <i>MyClasses::foo</i> , the <code>::</code> character indicating that the Class is within another namespace.	
Allow change of Created Date	Enable the creation date on the Diagram Properties dialog to be altered.	
Zoom to best scale	Resize diagrams to neatly fit the screen.	
Auto-pan with middle mouse button	Turn on auto-panning using the middle mouse button. With this option off, the middle mouse button causes a different type of panning.	
Scale Saved Bitmaps to	Enable Enterprise Architect to save bitmaps at a higher <i>resolution</i> , suitable for using in published works.	
Image Memory Limit	Set an image memory limit when generating images for RTF or HTML and when saving images to file. It is important when you have very large diagrams, as it affects the point at which Enterprise Architect starts to scale down the image; a low memory setting means it scales the image sooner.	
Diagram Frames	Select whether diagram frames are to be automatically added to images of diagrams in files saved to disk, print-outs, and the default Enterprise Architect clipboard. A diagram frame is a labeled outline around the diagram image, providing both a border and a reference.	Diagram Frame ^[888]

3.8.3.1 Appearance

The Diagram **Appearance** page of the **Options** dialog enables you to define how diagrams and their contents appear on display.

Access: **Tools | Options > Appearance**

Use to:

- Configure the appearance settings of a diagram and elements

Reference:

Field	Usage	See also
Configure Default Element Fonts	Set the default model and user text fonts.	Set default fonts ^[429]
Renderer	Render smooth curves and diagonal lines in diagrams, so that staggered vertical or horizontal pixels are less obvious. Select the type of renderer you prefer to use: <ul style="list-style-type: none"> • Basic is GDI32; it does not provide anti-aliasing and gradient fills • Enhanced-1 is parallel to Windows GDI+ but internal to Enterprise Architect; it provides anti-aliasing and gradient fills, 	

Field	Usage	See also
	<p>and operates well across different platforms</p> <ul style="list-style-type: none"> • Enhanced-2 is Windows GDI+; this can vary across different platforms, performing better than Enhanced-1 in some environments, and less well in others <p>Experiment with these options and see which works best for your system and requirements.</p>	
Default Element Border Width	Set the default element border width (in pixels).	
Scale view by	<p>Automatically increase the size of all objects on a diagram by up to 50%, without affecting other users reading that diagram.</p> <p>You can perform the same function with the Zoom Slider on the Status bar; changes in the 'zoomed' display scale of a diagram update this field and affect any other diagrams that you open.</p> <p>This has no impact any other diagram Zoom facility in Enterprise Architect.</p>	Status bar ¹¹⁷
Print in Color	Print your diagrams in color. Deselect the checkbox to print the diagrams in black and white.	
Anti-aliased text	<p>Force text anti-aliasing in diagrams.</p> <p>If you deselect the checkbox, Enterprise Architect applies the MS Windows default setting. Therefore, if you do not want to use anti-aliasing, ensure that the Windows anti-aliasing default is also set to OFF.</p>	
Compress text to fit within Element	Determine the behavior of Enterprise Architect when text at <i>zoom</i> levels other than 100% would not fit inside the boundary of an element. Enterprise Architect either compresses the text to fit within the boundary, or expands the element.	
Bold Element Names	Display element names in bold text.	
Element Shadows On	Display a shadow around the bottom and right edges of each element in a diagram.	
Invert rotated text for metafiles	Use different text format when external metafile readers are causing issues.	
Use Watermark	Add a watermark to any diagrams you print.	
Text	Define the watermark text, if a watermark is to be used.	
Show Gradient Fill for Paper Color	Switch between having a color gradient in the diagram background, or having a solid, uniform background color.	
Gradient Fill Direction For an Element	Select the direction for the color gradient within element boxes, or <none> for no color gradient.	

3.8.3.1.1 Set Default Fonts

Enterprise Architect enables you to define a standard font to apply across the model, or a font to apply to any diagrams you create personally.

You can define both, but the model font overrides any user font, to ensure that all members of a project team have a consistent and coherent view of the model. This avoids the problem of one user creating a diagram in a small font, and another user trying to view it in a larger font, which distorts the diagram.

It is recommended that a project authority sets the model default, and all project members abide by it and do not change it without project approval.

Access: **Tools | Options > Appearance : Configure Default Fonts**

Use to:

- Configure Font/s as the default for the model
- Configure Font/s as the default for diagrams created by the user

Reference:

Field	Usage	See also
Model Font : Font Face	Specifies the font face used by the overall model.	
Model Font : Font Size	Specifies the font size used by the overall model.	
Clear	Clears the selections made for Font Face and Font Size.	
User Font: Font Face	Specifies the Font Face used on diagrams if no model font is specified.	
User Font: Font Size	Specifies the Font Size used on diagrams if no model font size is specified.	

Notes:

- Both model and user fonts are overridden by specifically-defined element fonts, so that the element is viewed as designed regardless of the model or user defaults. To define the font for a specific element, right-click on the element in a diagram and select the **Appearance | Set Font** context menu option
- If you cannot read the diagrams because the default font makes the objects and text too small, you can scale up all objects (that is, all diagram displays) to a more readable size for you only; the objects are not scaled up for other users. Everything on the diagram is enlarged to the same extent, so it remains in proportion and readable. To do this, return to the **Diagram Appearance** page of the Options dialog and enter a suitable percentage value in the Scale view by field

Learn More:

- [Set Element Font](#)^[65]
- [Appearance](#)^[42]

3.8.3.2 Behavior

The **Diagram Behavior** page of the **Options** dialog enables you to define how a diagram responds to actions taken on it.

Access: **Tools | Options > Diagram : Behaviour**

Use to:

- Define diagram behaviour

Reference:

Field	Usage	See also
Auto Instance	Automatically create object instances when dragging certain element types - such as Class and Component - from the Project Browser, with the dragged element as the classifier	
Instance has Classifier style	Automatically apply the classifier style of the element an instance is instantiated from when the instance is created	
Show Life Lines	Show life lines for Sequence elements in non-Sequence diagrams	
Layout uses all relations	Show all relationships in a diagram layout; deselect the checkbox to show only Generalizations and Associations	
Auto Save Changes	Automatically save your changes as you work, without having to confirm prompts to do so	
Show 'Hidden' Parents	Display any parents of elements in the diagram that are not part of the diagram	
Auto Group Elements	Also move visually composed elements when moving diagram nodes; a node is considered composed if it is contained by the moved element and has a higher Z-order Press and hold (Alt) whilst moving an element to toggle this option	Diagram Context Menu ^[538]
Show Linked Items in Package	Display connected items on packages	
Show Package Item Scope	Display the + and - indicators representing the scope of the items	
Shift-Mouse Autoscroll	Enable you to press and hold (Shift) and use the mouse to autoscroll around diagrams	
Use Automatic SubActivities	Generate a new Structured Activity linked to the diagram from a Structured Activity diagram dragged from the Project Browser	
Always Highlight Context Element	Show a hatch border around a selected element	Highlight Context Element ^[645]

Field	Usage	See also
Objects Snap to Grid	Snap all elements to the grid lines	
Show Grid	Display the grid	
Grid Size	Specify the grid size, if you have selected Objects Snap to Grid	
Auto Tidy	Automatically tidy line angles for custom connectors; this 'nudges' the custom line into horizontal and vertical increments	Connector Styles [748]
Tidy line gap	Specify the amount Enterprise Architect should enable you to move a line away from horizontal and vertical when you are tidying lines for custom connectors. (See Auto Tidy above)	Connector Styles [748]
Alias only	Display the alias instead of the element name on elements with aliases	
Alias and Name	Display both the element name and the Alias in the format (<i>Alias</i>) <i>name</i>	
Use alias field for role	Replace the Alias property of instances with a Role property	

3.8.3.3 Sequence

The **Sequence** page of the **Options** dialog enables you to configure various font settings and the focus of the control indicator for Sequence diagrams.

Access: **Tools | Options > Diagram : Sequence**

Use to:

- Configure font settings
- Configure the focus of the control indicator for Sequence diagrams

Reference:

Field	Usage	See also
Message Spacing	Specify the vertical gap (in points) between Sequence messages (can be overridden manually by dragging a message up or down).	
Default page layout is Landscape	Set the default orientation of Sequence diagrams to landscape.	
Show Sequence Numbering	Show sequence numbers on Sequence messages.	
Assume right to left messages are Return	Automatically generate return messages.	
Default concurrency is Asynchronous	Set the default concurrency for Sequence Messages to Asynchronous ; deselect to set the default concurrency to Synchronous .	Message (Sequence Diagram) [987]

Field	Usage	See also
Width	Select the line width (in points) of the 'focus of control' rectangle (thick part of lifeline).	
Assume message returns	Assume implicit returns when none are explicitly drawn (recommended).	
GarbageCollect	Automatically truncate lifelines for created elements after the last message (that is, assume garbage collect rather than explicit delete).	
Name	Display the MS Windows Font dialog click on (...) and define the font of the caption bar heading (above your diagram); this is particularly useful for non-English character sets.	
Size	Specify the size of the heading font (this overrides the font size in the Font dialog, above).	
Use System	Apply the Enterprise Architect system default heading font.	

3.8.4 Objects

The **Objects** page of the **Options** dialog enables you to configure how elements look and respond in diagrams.

Access: **Tools | Options > Objects**

Use to:

- Change how elements look and respond

Reference:

Field	Usage	See also
Version	Set the default version for new elements.	
Phase	Set the default phase for new elements.	
Highlight References	Highlight parameters in operations that are passed by reference rather than value.	
Reference Char(s)	Specify a character to use for the reference.	
Prefix/Suffix	Indicate whether to use the Reference Char(s) value as a prefix (before) or a suffix (after).	
Warn about spaces in class names	Enable or hide the warning message that a Class, operation or attribute name has embedded spaces (which can cause coding problems).	
Classes honor analysis stereotypes	Show Classes as their stereotype; for example, if a Class is stereotyped as a Boundary, it appears as a Boundary rather than a Class.	
Show stereotype icon for requirements	Show or hide a code letter in the top right corner of Requirement (E, for external), Change (C) and Issue (I)	

Field	Usage	See also
	elements.	
Support for Composite Objects	Enable you to drag child elements onto parent elements in a diagram, and automatically embed them (and drag embedded child elements out of parent elements, breaking the child-parent relationship).	Move Elements Within Diagrams ^[632]
Auto-resize marks diagram 'dirty'	Ensure that auto-resizing of elements (such as Classes) marks the current diagram as changed (asterisk on the diagram name tab), so it should be saved.	
Highlight {abstract} elements	Highlight abstract elements with a suitable tag <i>{abstract}</i> in the top right of the Class.	
Allow elongated Use Cases	Stretch Use Cases or Use Case extension points with long names to enable space for the name. If you deselect the checkbox, Use Case re-sizing is proportional.	
Show status colors on diagrams	Enable color coding ^[1397] for Requirements and similar elements.	Status Types ^[783]
Copy inheritance links on duplicate	Duplicate Inheritance and Realization connectors when an edit/copy is performed.	
Port and Part type visible by default	Enable Port and Part types to be shown by default.	
Show buttons for selected Object on diagram	Display the floating toolbar buttons to the left of the selected object. For example: 	
Edit Object on New	Automatically show the element Properties dialog when a new element is added.	
Show «column» stereotype	Hide or show the « <i>column</i> » stereotype used when data modeling.	
Extend Complexity	Extend levels of complexity to five levels in the Complexity option in the Properties window. Otherwise only three levels are available.	
UML 1.5 Components	Use UML 1.5 components (Enterprise Architect versions 4.0 and later support UML 2.x).	
Show State Compartment	Show or hide the State Compartment divider under the state name.	
Show Duplicate Tags	Enable duplicate tags to be shown.	
Group Operations by Stereotype	Group an element's operations by their stereotype on the diagram.	
Group Attributes by Stereotype	Group an element's attributes by their stereotype on the diagram.	
Sort Features Alphabetically	Sort element features alphabetically. Features include Attributes, Operations, Tags, Constraints and Test Cases.	

Field	Usage	See also
Disable spelling	Turn off automatic spell checking. Deselect to resume automatic spell checking.	
Advanced	Set the visibility of certain elements in reports and in diagram packages	Element Visibility [434]

3.8.4.1 Element Visibility

Some elements do not appear in packages and in RTF output by default. Click on the **Advanced** button on the Objects page of the Options dialog to specify which elements should be visible.

Access: **Tools | Options > Objects : Advanced**

Use to:

- Specify which elements should be visible within RTF reports

Learn More:

- [Objects](#) [432]
- [Customize Visibility of Elements](#) [638]

3.8.5 Links

The **Links** page of the **Options** dialog provides options for the creation, behavior and notation for connectors.

Access: **Tools | Options > Links**

Use to:

- Provides options for connector behaviour

Reference:

Field	Usage	See also
Edit Connector on New	Automatically show the connector Properties dialog when a new connector is added.	
Association default = source --> target	Set the direction of new Associations to source->target (that is, with an arrow head at the target).	
Generalization link style Default = Tree	Show Generalizations as tree style hierarchies.	
Shade Qualifier boxes	Lightly shade all Qualifier boxes.	
Draw Aggregations	Draw Aggregate and Composite connectors from target element to source element. When deselected (the default),	

Field	Usage	See also
Reversed	these connectors are drawn from source to target.	
Prompt on connector deletes	Display a prompt before deleting connectors, offering the choice of hiding the connector on the diagram or deleting it completely. If you deselect this option, the delete operation defaults to the last setting on the dialog.	Delete Connectors [750]
Suppress Link Constraints	Suppress connector constraints in diagrams.	
Suppress Qualifier boxes	Suppress boxes when displaying qualifiers.	
Show Uses arrowheads	Show an arrowhead on Actor->Use Case Associations.	
Show Override Operation dialog on new connector	Show the Override Operation dialog automatically when adding generalizations and realizations between Classes and Interfaces, if the target element has features that can be overridden.	
Suppress ' + ' Role Scope	Ensure that the role and scope are not displayed on the diagram.	
Pen Width	Set the default connector width.	
Routing	Set the default connector style for new connectors.	
Enable	Enable the Quick Linker.	The Quick Linker [622]
Show Help	Add a 'help' menu option to the end of the Quick Linker menu.	
Center to center	Change the position of the dashed guide line for new connectors.	
Exact placement		
Force perpendicular line		

Notes:

- For **Draw Aggregations Reversed**, all tools have the parent as the target and the child as the source of the connector, that is a requirement of UML; only the direction in which you drag the mouse to draw the connector is changed

3.8.6 Communication Message Colors

The Communication Message Coloring page in the Options dialog enables you to configure the colors used in Communication diagrams.

Access: **Tools | Options > Communication Colors**

Use to:

- Configure colors used in Communication diagrams

Notes:

- Select the **Use Communication Message Coloring** checkbox to turn on colored messages. When you enable this option, Communication messages appear in different colors depending on the sequence group they belong to on a diagram; for example, 1.n are black, 2.n are red, 3.n are green
- Click on the down arrow in each color field, and click on the appropriate color for the message group. Set the color sequence as required; the pattern repeats after 8 sequence groups

3.8.7 XML Specifications

The XML Specifications page of the Options dialog enables you to configure various settings for working with XML.

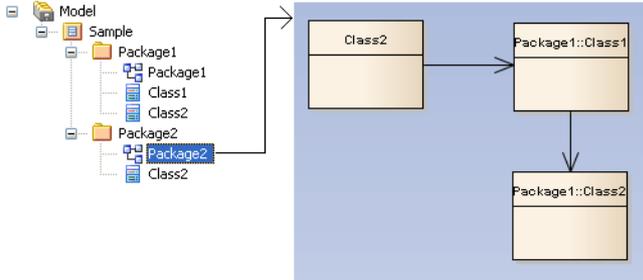
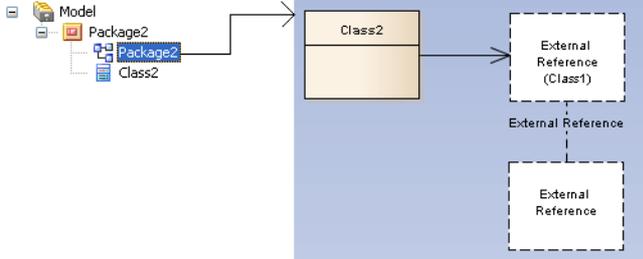
Access: **Tools | Options > XML Specifications**

Use to:

- Configure various XML Settings

Reference:

Field	Usage	See also
Editor	Set the default editor for any XML documents you open within Enterprise Architect	
Format XMI Output	Set whether or not formatting is applied to your XMI output	
Write Log	Set whether or not to write to a log file when you import or export XML	
Use DTD	Set whether or not to use a Data Type Definition	
Export Diagrams	Set whether or not to export diagrams when you export XML	
Export Alternate Images	Set whether or not to export the alternative images used in the model when you export to XML	Using the Image Manager ^[593] Check in a Package ^[285] Sharing Reference Data ^[236]
Default XMI Version	Set the XMI version to use: Enterprise Architect or Rose	
Code Page	Set the Code Page to use Setting a NULL encoding string results in the encoding tag being entirely omitted from the XML output Click on the Default button to restore the setting to the default Code Page.	
Export Diagram Images	Set whether or not to export diagrams as images when you export XML	
Image Type	Define the format of the image to export to if Export Diagram Images is selected	

Field	Usage	See also
<p>Create placeholders for missing External References during XMI 1.1/2.1 Import</p>	<p>Set this option to show, during an Enterprise Architect exported XML 1.1/2.1 file import, any missing:</p> <ul style="list-style-type: none"> • Diagram elements, using a Boundary element as a placeholder • Diagram connectors, using a Dependency connector as a placeholder <p>When importing an Enterprise Architect exported XML 1.1/2.1 file with cross-package references, use this option to visually show the diagram items that are not yet in the model</p> <p>A sample scenario would be exporting <i>Package2</i> to XML 1.1 from <i>Master.EAP</i> and importing it into <i>New.EAP</i> - the Classes <i>Class1</i> and <i>Class2</i> and the connector between them are shown using a placeholder on the diagram <i>Package2</i></p> <p>Master.EAP :</p>  <p>New.EAP :</p> 	
<p>Prefix EA Tagged Values in XMI 1.0 with ea\$</p>	<p>Set whether or not to prefix any Enterprise Architect Tagged Values within any XML 1.0 you create, with ea\$</p>	
<p>Update Last Save Time</p>	<p>Set whether to update the timestamp of the last time controlled packages were saved</p>	
<p>Report Cross Package Reference Deletions (XMI 1.1)</p>	<p>Set this option to report any package cross-references that would be deleted when a Controlled Package is exported to XML 1.1</p>	<p>Report Deletion of Cross-Package References ^[335]</p>
<p>Default XML Directory</p>	<p>Define the default XML directory to use when importing and exporting XML</p>	

Notes:

- If **Export Alternate Images** is set, and you have packages in your model under version control, then any alternative images used in those packages are also exported to the version control repository

when you check in the packages; in this case, you would only select the checkbox if the alternative images are subject to frequent change

Otherwise, do not select this option and instead use **Export/Import Reference Data** to manage alternative images

Learn More:

- [Export MOF to XML](#)^[1632]

Part

IV

4 Navigate, Search & Trace

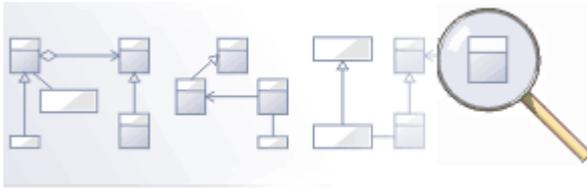


This section explains how you **navigate** through the model structures in Enterprise Architect, locate and display specific information, and trace the origins, development and fate of the model elements. While the **Project Browser** provides an effective hierarchical view of the model, large models can present particular challenges, and tracing **dependencies** and **relationships** is not always a simple task. Fortunately, Enterprise Architect has a number of tools designed to meet just these requirements.

Topics:

Topic	Link
The use of the Project Browser and Package Browser to navigate and explore your model	The Project Browser ^[441] Package Browser ^[456]
How Model Views provide different perspectives and 'entrypoints' into a model	Model Views ^[464] Diagram Slide Show ^[471]
Finding elements and relationships using the Model Search tools and custom queries	Model Search ^[475] Pre-defined Searches ^[479]
Traceability tools to help track completeness, dependencies and other behavioral and structural relationships	Traceability Tools ^[494]
Use of the Diagram List or Package Browser to view model elements in a conventional list based manner	Diagram List ^[462] Package Browser ^[456]
Use of the Relationship Matrix to understand and track relationships between model elements in a spreadsheet view	Relationship Matrix ^[496]
Using Diagram Filters to highlight information in diagrams	Diagram Filters ^[489]
Using the Traceability window to explore model dependencies	The Traceability Window ^[495]
Finding and working with Element Relationships	The Relationships Window ^[504]

4.1 Navigate: Exploring Your Model



Navigating is the process of systematically exploring the structure of your model using the hierarchical and list based views, such as the Project Browser. It is particularly useful when you are familiar with the structure of a model and its packaged contents. Whereas Searching provides a flat and random set of results, the Project and Package Browsers provide a highly structured viewpoint, reflecting the exact structure of your model.

Topics:

Topic	Link
Using the Project Browser, the main interface element for exploring and navigating your model	The Project Browser ^[441] Project Browser Context Menus ^[443] Project Browser Toolbar ^[452]
The Package Browser - a tabular, editable view of elements in a selected package, which can be displayed in the main workspace	Package Browser ^[456]
The Diagram List - a tabular, editable view of elements in a selected diagram, which can be displayed in the main workspace	Diagram List ^[462]
Using Model Views to provide different perspectives and 'entry points' into your model	Model Views ^[464]
Setting up a Diagram Slide show	Diagram Slide Show ^[471]
Using the Pan and Zoom window to navigate around very large diagrams	The Pan & Zoom Window ^[473]

4.1.1 The Project Browser

The Project Browser enables you to navigate through the Enterprise Architect project space. It displays packages, diagrams, elements and element features in a tree like structure, reflecting the arrangement of elements and packages within your model. The Project Browser is the primary mechanism for browsing and exploring your model and is the jumping off point for many of the most important features in Enterprise Architect.

Access: **View | Project Browser (Alt + 0)**

Use to:

- Navigate your model
- Drill down to specific elements
- Drag and drop elements and packages within the model
- Copy (duplicate) whole packages
- Import/export model information

- Set up version control, locking, execution analysis and many other aspects of the model tied to a particular package
- Document models
- Import code, xmi and csv files, database schema and other external data sources

If you right-click on an item in the Project Browser to display the context menus, you can perform additional actions such as adding new packages, creating diagrams, renaming items, creating documentation and other reports, and deleting model elements. You can also edit the name of any item in the Project Browser by selecting the item and pressing (**F2**).

Views

The Project Browser can be divided into Views, each of which contains diagrams, packages and other elements. A default View hierarchy is described below, but you can create different Views to suit your requirements:

View	Description
Use Case View	The functional and early analysis View. Contains Business Process and Use Case models.
Dynamic View	Contains State Charts, Activity and Interaction diagrams. The dynamics of your system.
Logical View	The Class Model and Domain Model View.
Component View	A View for your system components. The high level view of what software is to be built (such as executables, DLLs and components).
Deployment View	The physical model; what hardware is to be deployed and what software is to run on it.
Custom View	A work area for other Views, such as formal requirements, recycle bin, interview notes and non-functional requirements.

Selective Collapse:

When you are working on an expanded project in the Project Browser, you might want to locate the parent element or package of an item, and/or collapse the structure under that parent element or package.

How to:

To focus on the parent element or package, follow the steps below:

Step	Action	See also
1	Position the cursor on an item within the element or package	
2	Press (←) on the keyboard to highlight the parent.	
3	Press the key again to collapse the structure under that parent element or package.	

Learn More:

- [Views](#) ^[530]
- [Project Browser Icon Overlays](#) ^[454]

4.1.1.1 Project Browser Context Menus

This section describes the many options available to work on objects in the **Project Browser**, selected from the following context menus:

- [Model \(Root Node\) context menu](#) ^[443]
- [Package Context Menu](#) ^[444]
- [Element Context Menu](#) ^[448]
- [Diagram Context Menu](#) ^[451]
- [Operation Context Menu](#) ^[452]

4.1.1.1.1 Model (Root Node) Context Menu

The *Root Node* in the Project Browser is the *Model* element. You can have more than one model element.

The first level packages beneath the Model node are sometimes referred to as *Views* as they commonly divide a model into categories such as Use Case Model and Logical Model.

Right-click on the Root Node to display the **Model** context menu.

Action	Usage	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project		
Scripts	List the scripts enabled for execution directly from the Project Browser (Does not display if no Project Browser scripts exist)		
Package Control	Display the Package Control submenu		Package Control ^[328]
Rename Model	Rename the current model		
Add	Display the Add submenu; see below		
Copy Package to Clipboard	Copy the selected package to the clipboard, to be copied into another package in the same .eap file or a different .eap file		Copy the selected package ^[535]
Paste Package from Clipboard	Paste a package from the clipboard into the selected package		Paste a package ^[535]
Find in Project Browser	Find a specified term in the Project Browser	Ctrl + Shift + F	
Expand Branch	Expand all items		
Collapse Branch	Collapse all items		
Import Model from XMI	Import a model from an XMI file	Ctrl + Alt + I	Import a model ^[319]
Export Model to XMI	Export a model to XMI	Ctrl + Alt + E	Export a model ^[319]
Rich Text Format (RTF) Report	Produce RTF documentation for the model	F8	Produce RTF documentation

Action	Usage	Shortcut	See also
			[1726]
HTML Report	Produce HTML documentation for the model	Shift + F8	Produce HTML documentation [1802]
Diagrams Only Report	Produce a Diagrams Only report (in RTF) for the model	Ctrl + Shift + F8	Diagrams Only Report [1784]
Copy Reference	Copy a reference to the root node to the Enterprise Architect clipboard. Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> selected package hierarchy structure (node path) or node GUID 		
Delete Project Root	Delete the Model node and all subordinate Views and packages		
Help	Display the Help topic for the Project Browser		

Add Submenu:

Action	Usage	Shortcut	See also
Add Model (root node)	Create a new model root		
Add View	Create a new View (package)		
Add a New Model using Wizard	Add further models using the Model Wizard	Ctrl + Shift + M	Model Wizard [518]

4.1.1.1.2 Package Menu

Right-click on a View or Package in the Project Browser. The context menu displays, providing the following options:

Action	Usage	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project		
Scripts	List the scripts enabled for execution directly from the Project Browser (Does not display if no Project Browser scripts exist)		Script Group Properties [1819]
Properties	Display the Properties dialog for the Package element		Properties Dialog [660]
Package Control	Submit packages to package control and version control		Controlled Package Menu [328] Package Version Control Menu [278]

Action	Usage	Shortcut	See also
Add	Add a new diagram, element or another package to the current package		Add sub menu ^[446]
Paste Diagram	If you have copied a diagram from another package, paste the diagram into the currently-selected package		
Lock Package	Set a lock on the package		Lock Package ^[212]
Package Browser	Display the Package Browser, showing the elements contained in the selected package		Package Browser ^[456]
Find in All Diagrams	List all diagrams in which this package is depicted, in the Element Usage dialog	Ctrl + U	Show Element Use ^[630]
Turn On Level Numbering (Turn Off Level Numbering)	<p>Add a sequence number to each element in the package, based on the element's position in the package hierarchy</p> <p>For nested elements, the numbering indicates level; that is:</p> <p>3.2 3.2.1 3.2.1.1</p> <p>This option is only available for packages, and the numbering only applies to the elements in the package, not diagrams</p> <p>If elements are added, moved or deleted from the package, the numbering automatically adjusts</p>		Model Requirements ^[1165]
Linked Document	Create or display a linked document for the package or view	Ctrl + Alt + D	Linked documents ^[729]
Delete Linked Document	Delete the linked document attached to the package. The system prompts you to confirm the deletion		
Documentation	Produce a variety of reports and documentation in RTF format		Documentation Sub-menu ^[447]
Code Engineering	Perform Code Engineering functions		Code Engineering Sub-menu ^[447]
Execution Analyzer	<p>Display the Execution Analyzer window and highlight the Analyzer Scripts for the selected Package</p> <p>If no Analyzer Scripts have been configured you are given the opportunity to create one</p>		Visual Execution Analyzer ^[1635]
Import/Export	Import and export using XML text files		Import/Export Sub-menu ^[448]
Transform Current Package	Perform a model transformation on the selected package	Ctrl + Shift + H	Transform Elements ^[1305]
Contents	Reorganize the package contents after making		Contents Sub-

Action	Usage	Shortcut	See also
	changes		menu ^[448]
Bookmarks	Bookmark all elements in the selected folder		Manage Bookmarks ^[412]
Find in Project Browser	Search the Project Browser for specific elements (identified using the Find in Project Browser dialog)	Ctrl + Shift + F	
Copy Reference	Copy a reference to the package to the Enterprise Architect clipboard; select the appropriate sub-option to copy the: <ul style="list-style-type: none"> selected package hierarchy structure (node path) or node GUID 		
Copy Package to Clipboard	Copy the selected package to the clipboard, to be copied into another package in the same .eap file or a different .eap file		Copy a Package ^[535]
Paste Package from Clipboard	Paste a package from the clipboard into the selected package		Copy a Package ^[535]
Paste Element(s) from Clipboard	Paste elements copied to the clipboard into the selected package		Copy Elements Between Packages ^[634]
Save Package as UML Profile	Save the selected package as a Profile		UML Profiles ^[1026]
Set View Icon	Change the display icon for the selected package (View level)		
Move up	Move the package up in the list		
Move down	Move the package down the list		
Delete <packagename>	Delete the selected package and its contents		
Help	Display the Help topic for the Project Browser		

4.1.1.1.2.1 Add Sub-Menu

In the Project Browser, right-click on a package and select the **Add** context menu option.

Action	Shortcut	See also
Add Diagram		New Diagram ^[568]
Add Element	Ctrl + M	New Element ^[627]
Add Package	Ctrl + W	Add Package ^[534]
Add a New Model using Wizard		Model Wizard ^[518]

4.1.1.1.2.2 Documentation Sub-Menu

In the Project Browser, right-click on a package and select the **Documentation** context menu option.

Action	Shortcut	See also
Rich Text Format (RTF) Report	F8	RTF Document ^[1726]
HTML Report	Shift + F8	HTML Report ^[1802]
Diagrams Only Report	Ctrl + Shift + F8	Diagrams Only Report ^[1784]
Testing Report		Test Document ^[1708]
Open in Relationship Matrix		Element Relationship Matrix ^[496]
Generated Report Options		RTF Report Options ^[1731]
Copy RTF Bookmark		Copy RTF Bookmark ^[1796]
Package Metrics		Package Metrics ^[404]

4.1.1.1.2.3 Code Engineering Sub-Menu

In the **Project Browser**, right-click on a package and select the **Code Engineering** context menu option.

Action	Shortcut	See also
Generate Source Code	Ctrl + Alt + K	Generate Source Code ^[1490]
Import Source Directory	Ctrl + Shift + U	Import Source Directory ^[1513]
Import Binary Module		Import Binary ^[1513]
Synchronize Package With Code	Ctrl + Alt + M	Synchronize Package With Code ^[1506]
Generate DDL		Generate DDL ^[1380]
Import DB schema from ODBC		Import DB schema from ODBC ^[1375]
Generate XML Schema		Generate XSD ^[1598]
Import XML Schema		Import XSD ^[1600]
Generate WSDL		Generate WSDL ^[1621]
Import WSDL		Import WSDL ^[1622]
Reset Options for this Package		Reset Options ^[1540]
Reset DBMS Options		Reset DBMS Options ^[1356]
Set as Namespace Root/ Clear Namespace Root		Namespaces ^[1495]

Action	Shortcut	See also
Live Code Generation - Toggle between using and not using live code generation; that is, to update your source code instantly as you make changes to your model		

4.1.1.1.2.4 Execution Analyzer Sub-Menu

In the Project Browser, right-click on a package and select the [Execution Analyzer](#)^[1635] context menu option. This menu option displays the Execution Analyzer window and highlights the Analyzer Scripts for the selected Package. If no any Analyzer Scripts have been configured you are given the opportunity to create one.

4.1.1.1.2.5 Import/Export Sub-Menu

In the **Project Browser**, right-click on a package and select the **Import/Export** context menu option.

Action	Shortcut	See also
Import package from XMI file	Ctrl + Alt + I	Import package ^[323]
Export package to XMI file	Ctrl + Alt + E	Export package ^[320]
CSV Import / Export		CSV Import ^[342] / Export ^[340]

4.1.1.1.2.6 Contents Sub-Menu

In the Project Browser, right-click on a package and select the **Contents** context menu option.

Menu Option	Action	See also
Expand Branch	Expand all of the items in the Project Browser .	
Collapse Branch	Collapse all of the items in the Project Browser .	
Reset Sort Order	Return sorting of package contents to list in alphabetical order.	
Reload current package	Refresh the current package in the Project Browser .	Refresh Model View ^[187]

4.1.1.1.3 Element Menu - Project Browser

Right-click on an *element* (such as Class, Object, Activity, State) in the Project Browser to display the element's context menu.

Command	Action	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project		
Scripts	List the scripts enabled for execution directly from the Project Browser		Scripts ^[1819]

Command	Action	Shortcut	See also
	(Does not display if no Project Browser scripts exist)		
Properties	View and modify the element properties		
Add	Create a child element and diagram (Classifier elements) or a connector to another element		Add sub menu ^[450]
Paste Diagram	Pastes a copy of a diagram from the clipboard as a child diagram of the selected element The Copy Diagram dialog displays, offering the choice between a shallow copy and a deep copy		Copy (Duplicate) Diagram ^[583]
Rule Composer	For a Rule Task element, invoke the Rule Composer tab in Business Rule Modeling		Rule Composer ^[121] tab
Attributes	Display the Attribute dialog ready to create a new attribute		
Operations	Display the Operations dialog ready to create a new operation		
Create Workbench Instance	Create workbench variables from the selected Class When you select this option, Enterprise Architect prompts you to name the variable, which then displays in the Workbench window	Ctrl + Shift + J	Create workbench variables ^[166] Workbench ^[166] window
Generate Code	Generate the source code for this element	F11	Generate Source Code ^[149]
Synchronize with Code	Synchronize the element in the diagram with the source code	F7	Import Source Code ^[150]
View Source Code	View the source code of the element	F12	The Source Code Viewer ^[140]
Open Source Directory	Open the source directory	Ctrl + Alt + Y	
Find In All Diagrams	Locate the element in all open diagrams	Ctrl + U	
Locate in Current Diagram	Select the element in the currently-visible diagram If the element is not in the diagram, this option is grayed-out		
Copy RTF Bookmark	Copy a bookmark in RTF format to the clipboard		Bookmark ^[179]
Linked Document	Create a Linked Document (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions)	Ctrl + Alt + D	Linked Documents ^[72]
Delete Linked Document	Delete the linked document attached to the selected element		
Add Custom	Set up cross references between elements in a	Ctrl + J	Cross references

Command	Action	Shortcut	See also
Reference	diagram and the selected element in the Project Browser		[630]
Copy Element(s) to Clipboard	Copy the selected element or elements to the clipboard to be pasted into another package in this .eap file or another .eap file		Copy the selected element [634]
Copy Reference	Copy a reference to the element to the Enterprise Architect clipboard. Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> selected element hierarchy structure (node path) or node GUID 		
Move Up	Move the element up in the list of elements within this package		
Move Down	Move the element down in the list of elements within this package		
Delete '<element Name>'	Delete the element		
Help	Display the Help topic for the Project Browser		

4.1.1.1.3.1 Add Sub Menu

Access: **Right-click element in Project Browser: (Insert)**
Right-click element | Add

Menu Option	Action	See also
Activity	Add an Activity behavior element and one of its associated diagrams to the selected Classifier element	Activity [873] Object Classifiers [705]
Interaction	Add an Interaction behavior element and one of its associated diagrams to the selected Classifier element	Interaction Occurrence [908] Object Classifiers [705]
State Machine	Add a State Machine Behavior element and one of its associated diagrams to the selected Classifier element	State Machine [928] Object Classifiers [705]
RuleFlow Activity	Add a Rule Flow Activity element and associated diagram to a Class, in Business Rule Modeling	Create a Rule Flow Model [1206]
Add Diagram	Create a diagram to explain or expand on the selected Classifier element, using the New Diagram dialog	Add New Diagrams [568]
Create Link	Create a connector to another element	Create Connector in Project Browser [748]

Notes:

- Elements such as Actors, Classes and Activities can define a large amount of information that can be conveniently represented by or expanded in a child diagram; the **Add** sub-menu for these elements

provides all of the options listed above

- Elements such as Timing, Exit and History have much more specific functions that do not require expansion; the **Add** sub-menu for these elements only provides the **Create Link** option

4.1.1.1.4 Diagram Menu - Project Browser

Right-click on a diagram in the **Project Browser** to open the **Diagram** context menu. The example below illustrates the functions available from this menu.

Action	Usage	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project.		
Scripts	List the scripts enabled for execution directly from the Project Browser . (Does not display if no Project Browser scripts exist.)		Scripts ^[1819]
Properties	View and modify a diagram's properties.	F5	
Open	Open the diagram in the Diagram View .		
View Diagram As List	Display the Diagram List, listing the elements in the selected diagram.		Diagram List ^[462]
Copy Diagram	Copy the diagram for pasting into another location.		Copy a Diagram ^[583]
Copy RTF Bookmark	Copy a bookmark in RTF format to the clipboard.		Bookmark ^[1796]
Add Custom Reference	Add this diagram as a cross reference to other elements.		Cross reference ^[630]
Print Diagram(s)	Print the currently-selected diagram or diagrams (hold (Ctrl) or (Shift) while selecting).		
Copy Reference	Copy a reference to the diagram node to the Enterprise Architect clipboard. Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> • selected hierarchy structure (node path) or • node GUID. 		
Move up	Move the diagram up in the list of diagrams within this package.		
Move down	Move the diagram down in the list of diagrams within this package.		
Delete '<diagram name>'	Delete the selected diagram.		
Delete selected items	Delete several selected diagrams (hold (Ctrl) or (Shift) while selecting).		
Help	Display the Help topic for the Project Browser .		

4.1.1.1.5 Operation Menu - Project Browser

To display the **Operation** (or Method) context menu, right-click on an Operation in the **Project Browser**.

Action	Usage	Shortcut	See also
Add-In	Access the facilities of each Add-In currently enabled for the project		
Scripts	List the scripts enabled for execution directly from the Project Browser (Does not display if no Project Browser scripts exist.)		Script Group Properties [1819]
Generate Code	Generate code for the operation	F11	
Synchronize With Code	Synchronize the operation with the code	F7	
View Source Code	Open the Source Code Viewer and display the operation	F12	The Source Code Viewer [1408]
Operation Properties	Display the Properties dialog for the operation		
Copy Reference	Copy a reference to the operation to the Enterprise Architect clipboard Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> selected hierarchy structure (node path) node GUID 		
Delete Operation	Delete the operation		
Help	Display the Help topic for the Project Browser		

Notes:

- You can display an equivalent context menu for an attribute by right-clicking on the attribute in the Project Browser.

4.1.1.2 Project Browser Toolbar

The Project Browser toolbar enables you to perform a range of operations on your project structure.

Access: **View | Project Browser**

Use to:

- Perform a variety of operations on the project structure and components

Reference:



Action	Usage	Shortcut	See also
New Model from Pattern	Create a new Model Package in the project, from a predefined UML or Technology pattern		Model Wizard ^[518]
Add a Package	Create a new child package under the selected package		Add a Package ^[534]
New Diagram	Create a new child diagram under the selected package or element		Add New Diagrams ^[568]
Create Element	Create a new child element under the selected package or element		Add Elements Directly to Packages ^[627]
Find in Project Browser	Perform a simple search for a text string in the Project Browser		
Documentation	Provide options to generate an: <ul style="list-style-type: none"> • RTF report • HTML report or • Diagram Only report on the selected package in the Project Browser	(F8) (Shift+F8) (Ctrl+Shift+F8)	Generate RTF Documents ^[1727] HTML Reports ^[1802] Diagrams Only Report ^[1784]
Code and DB Generation	Provide options to: <ul style="list-style-type: none"> • Generate source code • Generate DDL • Import a source directory • Import a binary module • Import a database schema • Generate package contents to synchronize with package code • Reset the source code language All for the selected package		Generate a Package ^[1493] Generate DDL for a Package ^[1380] import a Directory Structure ^[1513] Import Binary Module ^[1513] Import Database Schema ^[1375] Update Package Contents ^[1494] Reset Options ^[1540]
Move Element Up	Move the selected package or element further up the Project Browser, within its parent package		
Move Element Down	Move the selected package or element further down the Project Browser, within its parent package		
Help	Open the Enterprise Architect Help on the Project Browser.		

4.1.1.3 Project Browser Icon Overlays

The **Project Browser** displays the status of each package in the model by overlaying status icons on the package icon. The following table describes what each overlaid icon means.

Icon Overlay	Indicates that...	See also
	This package is controlled and is represented by an XMI file on disk. Version control either is not being used or is not available. You can edit the package.	Controlled Packages ^[327]
	This package is version controlled and checked out to you, therefore you can edit the package.	Use Version Control ^[275]
	This package is version controlled and not checked out to you, therefore you cannot edit the package (unless you check the package out).	Use Version Control ^[275]
	This package is version controlled, but you checked it out whilst not connected to the version control server. You can edit the package but there could be version conflicts when you check the package in again.	Offline Version Control ^[250]
	This package is a namespace root. It denotes where the namespace structure starts; packages below this point are generated as namespaces to code.	Namespaces ^[1495]
<MDG Add-In icon>	MDG Add-Ins specify their own icon to denote that this branch of the model belongs to that Add-In. All packages connected to an MDG Add-In correspond to a namespace root, so the namespace root icon is not displayed.	MDG Add-Ins ^[2042]

Similarly, the **Project Browser** indicates attribute and operation scope status with icons. The following table describes what each indicator icon means.

Icon Overlay	Indicates that...	See also
 	The attribute or operation is scoped as protected.	
 	The attribute or operation is scoped as private.	

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, if project User Security is on, the Project Browser also has element locking indicators (red and blue exclamation marks) that indicate the lock status of individual elements and packages

The availability of these elements for editing depends on whether user locks are required or not

Learn More:

- [Configure User Security](#)^[194]
- [Locked Element Indicators](#)^[214]

4.1.1.4 Order Package Contents

Ordering elements is very important when it comes to structuring your model, especially packages. RTF documents honor any custom ordering when printing documentation. Enterprise Architect enables you to change the order of elements listed in the Project Browser.

By default, elements are first listed in:

- order of type, then
- order of set position, then
- alphabetically

You can use the context menu options to move an element up or down within its type, but not outside its type. This means you can re-sequence Packages or Diagrams or Use Cases, but you cannot mix up elements.

However, you can change this default behavior to allow elements to be re-ordered within the package, regardless of type.

Learn More:

- [Set Default Behavior](#)^[455]

4.1.1.5 Set Default Behavior

The General page of the Options dialog provides several options for altering the look and behavior of the Project Browser.

Access: **Tools | Options > General**

Panel	Option	Effect	See also
Double-click On Browser	Shows Properties	When an item in the Project Browser is double-clicked, the item's Properties dialog (if available) displays	
	Opens Branch	When an item in the Project Browser is double-clicked, the hierarchy expands to show the item's children If there are no children, nothing happens	
	Opens Branch & Diagram	As above, but also opens the first diagram beneath the item, if applicable	
Project Browser	Allow Free Sorting	Enables you to re-order elements within a package regardless of type, within the Project Browser	Order Package Contents ^[455]
	Show Stereotypes	When a stereotype is defined for an element, the stereotype name then displays in front of the element name You set the stereotype of an element in its Properties dialog Shut down and restart Enterprise Architect to enable this change to take	

Panel	Option	Effect	See also
		effect	

4.1.2 Package Browser

The **Package Browser** is a tabular, editable view of elements in a selected package, which can be displayed in the main workspace.

Access: **Project Browser > package | View | Package Browser (Ctrl+Alt+R)**
Project Browser > Right-click package | Package Browser
Diagram > Right-click package | Package Browser

Use to:

- Streamline the process of creating and updating elements in a package selected from the Project Browser; this can be particularly useful for analysts to create and maintain formal requirement definitions within the model
- Print the list or generate an RTF document directly from the entries on the list

Topics:

Topic	Detail	See also
The Package Browser	The Package Browser is related to the Diagram List, which lists the elements in a selected diagram; however, the Package Browser has a wider range of options	Diagram List ^[462]
Formats	<p>The Package Browser has two formats:</p> <ul style="list-style-type: none"> • <i>User Defined</i> format, where you can change how the element information is structured on the screen using the value-grouping band below the toolbar • <i>Model Hierarchy</i> format, where the package and element hierarchies are represented in the display <p>Switch between these formats using the Show Hierarchy button () in the toolbar</p>	List Header ^[458]
Change display	<ul style="list-style-type: none"> • Sort the items by any column value in ascending or descending order, by clicking on the column header; initially the elements are listed in numerical order (if level numbering is turned on in the Project Browser) or alphabetical order within type • Apply a diagram filter for elements (the Fade, Gray Scale and Hide modes reduce the number of elements listed; Select mode has no effect) • Change the sequence of columns, by dragging column headers left or right • In <i>Model Hierarchy</i> format, the Name column is always on the left; you cannot move any other column into that position, although you can rearrange the order of the rest of the columns <p>Because of this, if you group or sort information in <i>User Defined</i> format and switch to <i>Model Hierarchy</i> format your information structure is altered, and it is not restored when you switch back to</p>	Diagram filters ^[489]

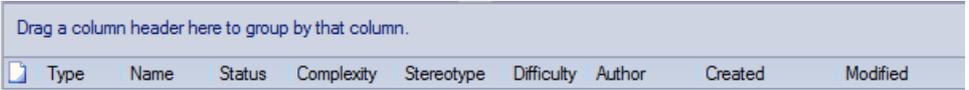
Topic	Detail	See also
	<i>User Defined</i> format	
Display Element Properties	Double-click on the element entry	
Show Elements in a Child Package	Double-click on the child package in the list	
Select Items	Select: <ul style="list-style-type: none"> • an element by clicking on it • a specific value by clicking twice on it (not double-clicking); either the value becomes directly editable or the Properties dialog displays in which you can edit the value • several individual elements by holding (Ctrl) as you click on them • a range of elements by holding (Shift) as you click on the first and last in the range 	
Add new items to the package	Click on a listed element and press (Ctrl+N) or (Insert), or right-click and select the Add New context menu option <ul style="list-style-type: none"> • in <i>Model Hierarchy</i> format, new elements are inserted in the order in which they appear in the Project Browser • in <i>User Defined</i> format, new items are inserted to comply with any sort order and/or grouping; if the list is not sorted or grouped, the items are added to the end 	
Add a child element to the selected element <i>(Model Hierarchy format)</i>	Press (Esc) and then (Ctrl+N), otherwise, elements are added as siblings of the selected element You can add child elements only when the whole row is selected, with none of the row cells or fields highlighted; press (Esc) to remove selection from an individual cell	
Add elements to the Package Browser	Drag them from anywhere in the Project Browser You can also drag elements <i>from</i> the Package Browser <i>into</i> the Project Browser	
Delete elements from the list	Select the item and press (Ctrl+D) In <i>Model Hierarchy</i> format, you cannot delete a parent element until all its child elements have been removed or deleted	
Include each element's notes (documentation)	Notes are shown underneath the element To add or edit notes, click on the item and press (Ctrl+Space) to transfer control to the Notes window, in which you create or edit the note text	Package Browser Options <small>[459]</small>
Review the element's custom (advanced) properties	Click on the item and press (Ctrl+Enter) The Custom Properties dialog for the element displays	

Topic	Detail	See also
Further Work	You can do further work on the Package Browser using the toolbar and context menu options	Package Browser Options ^[459]

4.1.2.1 List Header

This topic describes the organization of the column headings and contents for various reports in Enterprise Architect.

Topics:

Topic	Description
View Header	 <p>The View Header defines the columns of information that are presented by the report, and the order in which data items are presented</p> <p>To review the column headings:</p> <ul style="list-style-type: none"> Right-click on the header to display the Field Chooser context menu option; the Field Chooser dialog displays <p>This enables you to add columns to or remove columns from the output; between them, the View Header and Field Chooser dialog show the full range of column headers available</p> <p>To add a column heading to the View Header:</p> <ul style="list-style-type: none"> Drag it from the Field Chooser dialog onto the header to the position you want the column of data to display When you have selected the column headings you require, click on the red cross in the top right corner of the Field Chooser dialog to close it <p>To remove a column from the output:</p> <ul style="list-style-type: none"> Drag the column heading downwards out of the View Header <p>To change the sequence of columns:</p> <p>Drag the required column header left or right, as appropriate</p>
Value Grouping	<p>For most reports, and when the Package Browser or Diagram List is in <i>User Defined</i> format, you can organize the reported data according to the value of one or more of the column categories (this facility is not available in the Package Browser in <i>Model Hierarchy</i> format)</p> <p>You might organize the data by Type, and within each Type by Name; if you then click on any of the other column headings, the data within this grouping is further sorted with the values of the selected column (for example, Created) in ascending or descending order</p> <p>To set up the value grouping, drag the column heading representing the primary grouping (such as Author) onto the Drag a column header here to group by that column field</p> <p>Now drag the column heading for the next level of grouping (such as Status) to the right of the first heading; the two heading titles display as connected blocks, as shown below:</p>

Topic	Description
	 <p>You can, if required, add further levels of grouping by dragging other column headings onto the hierarchy (such as Created), and restructure the order by dragging existing or additional headings into the level you want them to hold</p> <p>For example, you could make Type the secondary grouping by dragging it to the right of Author, or drop Status between Author and Type</p> <p>To remove a grouping level, drag the appropriate column heading out of the sequence and below the View header; any subordinate groupings move up a level</p>
<p>Filter Bar</p>	<p>The Filter Bar displays underneath the column headings; you can hide or show it using the Toggle Filter Bar icon in the toolbar</p> <p>The Filter Bar consists of a row of fields, one field per column; as you type a value into one of these fields, the Package Browser results are automatically filtered to show entries with a value in the corresponding column that match the value in the field. The filter is case sensitive.</p> <p>For example, if you type the letter F in the Filter Bar field under the Author column, the results immediately reduce to entries for which the author name includes Fred, Frank, Felix, Felicity, Fairmile, Farmer or Fitzgibbon (but none where the name includes f only as a lower case letter, such as Rufus)</p> <p>If you then go on to type Fi, the results immediately reduce again to only entries for which the author name includes Fitzgibbon.</p>

Learn More:

- [Package Browser](#)^[456]
- [Diagram List](#)^[462]
- [Report View](#)^[370]

4.1.2.2 Package Browser Options

You can add to or influence what information is displayed on the Package Browser by selecting the window's toolbar or context menu options.

Toolbar:

Icon	Action	Shortcut	Link
	<p>Add a new element to the package</p>	<p>Ctrl + N</p>	<p>Add Elements Directly to Packages^[627]</p>
	<p>Display the Notes window, to add or edit notes for the selected element</p>	<p>Ctrl + Space</p>	
	<p>Delete the selected element</p>	<p>Ctrl + D</p>	
	<p>Print the current contents of the Package Browser</p>		

	Display the Generate RTF Documentation dialog, to create one RTF report on all selected element(s)		Generate RTF Documentation Dialog ^[1732]
	Select: <ul style="list-style-type: none"> The appropriate element type from the drop-down list, or All to list all objects The report then lists only elements of that specific type		
	Select a UML, Extended or MDG Toolbox category to specify the category of elements shown in the filter list (above)		
	Toggle between: <ul style="list-style-type: none"> Including child packages and their contents in the list, and Showing only the first-level contents of the selected diagram or package 		
	Toggle the display between: <ul style="list-style-type: none"> Model Hierarchy format and User-Defined, value-grouping format 		List Header ^[458]
	Display a short menu from which you can select to: <ul style="list-style-type: none"> Hide the contents of the Notes compartment of each element Display the first few words, or Display the full text 		
	Change the display to show the contents of the parent package of the selected package		
	Display Help on the Package Browser		

Context Menu Options:

Action	Usage	Shortcut	See also
Properties	Display the Properties dialog for the selected element		
Edit Notes	Add or edit notes on the element, in the Notes window	Ctrl + Space	
Add New	If the Filter List field in the toolbar is set to: <ul style="list-style-type: none"> All, display the New Element dialog, through which you create an element of the required type A specific element type, add an element of that type to the package or diagram in the Package Browser and the Project Browser 		
Find in Diagrams	Display: <ul style="list-style-type: none"> The diagram that uses the element or, 		Show Element Use ^[630]

Action	Usage	Shortcut	See also
	<ul style="list-style-type: none"> If the element is used in multiple diagrams, display the Element Usage dialog, which lists the diagrams that contain the element 		
Find In Project Browser	Highlight the selected element in the Project Browser		
Bookmark Item	Bookmark the element		
Create Linked Document (Edit Linked Document)	Create (or edit) a Linked Document (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions)	Ctrl + Alt + D	Linked Documents Link Document to UML Element Edit Linked Documents
Delete Linked Document	(If the element has a linked document) Delete an existing linked document		
Documentation	Generate an RTF report; you have two options: <ul style="list-style-type: none"> Generate a separate report on each selected object in the report Generate one report on all selected objects. In either case, the Generate RTF Documentation dialog displays You also have the option to print out the Package Browser list itself		Generate RTF Documentation Dialog
Sort Contents	In <i>Model Hierarchy</i> format, synchronize the list with the Project Browser hierarchy, to ensure that all element and package hierarchies and sequences are - if necessary - updated Normally changes are updated automatically, but there can be delays if changes are made outside the Package Browser		
Reload	Reload the list of elements to refresh the order and content with any recent changes		
Add Tag Value Column	Add a column to show the values for a selected Tagged Value, for each element in the list (see below)		
Print	Print the Package Browser		
Delete Selected	Delete the selected element from the Package Browser		

Add Tagged Value Column:

If you want to show the value that an element has for a particular tag (and the value that any other element in the list has for that tag), you can add a *read-only* column for the tag. There are two ways in which to do this:

Method	Action	See also
1	Open the Tagged Values window for the element and drag the tag onto the Package Browser; a new column is added for each tag you drag onto the list.	
2	<p>Right-click on the element in the Package Browser and select the Add Tag Value Column context menu option; the Add Tag Value Column dialog displays</p> <p>Either type the tag name into the Tag Value field, or select one of the two options:</p> <ul style="list-style-type: none"> • Global Tag Values - populates the Tag Value drop-down list with all global Tagged Values in the project; this enables you to select a Tagged Value that is applicable to a large number of elements • Tag Values from the selected Element - populates the Tag Value drop-down list with the Tagged Values defined for the selected element (as listed on the Tagged Values window for that element); this enables you to filter the initial selection to a specific set of tags <p>When you have selected the required tag, click on the OK button to insert a column for that tag into the Package Browser list.</p>	

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if Auditing is turned on and the Package Browser is open, you can view a history of changes to any selected element or connector, in the Audit History tab of the Output window (if security is enabled, you must have at least **Audit View** permissions to display the audit history)

Learn More:

- [Auditing](#) ^[298]
- [Audit History](#) ^[307] tab
- [List of Permissions](#) ^[205]

4.1.3 Diagram List

The Diagram List is a tabular, editable view of elements in a selected diagram, which can be displayed in the main workspace.

Access: **diagram | Diagram | Switch To | List View**
Project Browser > Right-click diagram | View Diagram As | List View
Project Browser > diagram (Ctrl+Alt+R)
diagram > Right-click | List View

Use to:

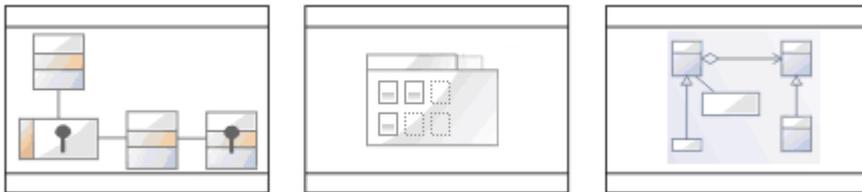
- Streamline the process of creating and updating elements in a diagram selected from the Project Browser; this can be particularly useful for analysts to create and maintain formal requirement definitions within the model
- Print the list or generate an RTF document directly from the entries on the list

Topics:

Topic	Detail	See also
The Diagram List	The Diagram List is related to the Package Browser, which lists the elements in a selected package; however, the Package Browser has a wider range of options	Package Browser ^[456]
Formats	The Diagram List is displayed in <i>User Defined</i> format, where you can change how the element information is structured on the screen using the value-grouping band below the toolbar	List Header ^[458]
Change display	<ul style="list-style-type: none"> Sort the items by any column value in ascending or descending order, by clicking on the column header; initially the elements are listed in numerical order (if level numbering is turned on in the Project Browser) or alphabetical order within type Apply a diagram filter for elements (the Fade, Gray Scale and Hide modes reduce the number of elements listed; Select mode has no effect) Change the sequence of columns, by dragging column headers left or right 	Diagram filters ^[489]
Display Element Properties	Double-click on the element entry	
Select Items	Select: <ul style="list-style-type: none"> an element by clicking on it a specific value by clicking twice on it (not double-clicking); either the value becomes directly editable or the Properties dialog displays in which you can edit the value several individual elements by holding (Ctrl) as you click on them a range of elements by holding (Shift) as you click on the first and last in the range 	
Add new items to the diagram	Create new elements on the Diagram List, using the toolbar or context menu options; these elements are automatically added to the diagram	Package Browser Options ^[459]
Delete elements from the list	Select the item and press (Ctrl+D)	
Include each element's notes (documentation)	Notes are shown underneath the element To add or edit notes, click on the item and press (Ctrl+Space) to transfer control to the Notes window, in which you create or edit the note text	Package Browser Options ^[459]
Review the element's custom (advanced) properties	Right-click on the item and select the Properties context menu option, then select the Advanced page	
Further Options	You can do further work on the Diagram List using the same toolbar and context menu options as for the Package Browser, except that the Diagram List <i>does not have</i> : <ul style="list-style-type: none"> The <i>Model Hierarchy</i> format and hence does not have the Show Hierarchy button () in the toolbar 	Package Browser Options ^[459] Diagram Properties

Topic	Detail	See also
	<ul style="list-style-type: none"> The  toggle button The  toggle button <p>The Diagram List <i>does have</i> these additional context menu options:</p> <ul style="list-style-type: none"> Diagram View - show the elements of the selected diagram as a diagram instead of as the Diagram List Gantt View - show the list as a Gantt chart of resource allocations Diagram Properties - display the Diagram Properties dialog for the diagram 	569

4.1.4 Model Views



The Model Views window enables you to encapsulate your model into the areas you are interested in.

Access: [View | Model Views](#)

Topics:

Topic	Detail	See Also
View Root Nodes	<p>There are three types of View root-node available:</p> <ul style="list-style-type: none"> <i>Model Views</i> - stored in the model and visible to all users; you can have many of these <i>My Views</i> - stored locally on your machine and visible only to you; you can have only one of these <i>Technology-defined Views</i> - read only; each View is stored with and populated by the corresponding active MDG Technology <p>Additionally, there is a <i>Recent Discussions</i> folder that contains current correspondence from the Team Review concerning items that are held in any of the Views; this folder has a separate repository of postings for each team review server connection you access through the model. You can control how recent these postings must be, and how many are to be listed.</p> <p>When you open the Model Views window for the first time on a project, a <i>Model Views</i> root section, <i>My Views</i> root section and <i>Recent Discussions</i> folder are added for you. These can not be deleted or renamed; however, you can create further Model View root nodes that you can modify and delete</p>	Model Views Operations 469 Team Review Tools 216 Team Review Connections 230 Model Views Context Menus 467
Subordinate Folders	<p>Under the <i>My Views</i> root node you can add a single level of <i>View</i> folders, which enable you to group <i>Search View</i> folders as best suit your requirements</p> <ul style="list-style-type: none"> A <i>Search View</i> is a folder of elements or structures that you 	Model Views Operations 469 Diagram Slide

Topic	Detail	See Also
	<p>assemble by assigning a model search to the folder; when you double-click on or expand the folder, the search runs and refreshes the folder contents</p> <ul style="list-style-type: none"> You can also set a search to refresh at a defined interval, and to notify you if new results are found <p>Under a <i>Model View</i> root node you can create Views folders that can contain Search View folders, <i>Slideshow</i> folders and <i>Favorites</i> folders</p> <ul style="list-style-type: none"> A <i>Slideshow folder</i> contains diagrams only, which you can display as a slide show with diagrams being shown in the sequence in which they are listed in the folder. One folder represents one slide show - you can run the slide show automatically or manually; in either case the diagrams are closed after they have been displayed. A <i>Favorites</i> folder gives you easy access to commonly-used items in the Project Browser; to create hyperlinks in a Favorites folder to the required items in the Project Browser, drag items from the Project Browser into the Favorites folder <p>You can also export all of the View folders containing Views from any root section as an XML file, and import a <i>Views XML file</i> as an additional, editable <i>Model View</i> root node</p> <p>These are single-level items; if you drag a package into the Favorites folder, you cannot expand that package there</p> <p>To select specific items inside a package, expand it in the Project Browser and then drag the items into the Favorites folder</p>	<p>Show ⁴⁷⁷</p>
<p>Key to Model View Contents</p>	<p>Items on the model view can be identified by their accompanying icons:</p> <ul style="list-style-type: none">  - A Model View root node  - A View Folder  - A Favorites Folder  - A View (search-based)  - A Slideshow Folder  - The My Views root node  - The DoDAF-MODAF View (a Technology-defined View)  - The Recent Discussions folder  - A repository of postings from a selected team review, concerning elements in the Model Views  - The recent posts folder for the selected team review 	
<p>Accessing Operations</p>	<p>To perform operations on the Model View contents, you can use the:</p> <ul style="list-style-type: none"> Model Views window Toolbar or the Item context menus <p>Each level of the Model Views hierarchy has a slightly different context menu.</p>	<p>Model Views Toolbar ⁴⁶⁶</p> <p>Model Views Context Menus ⁴⁶⁷</p>

4.1.4.1 Model Views Toolbar

Access: View | Model Views > Toolbar

Reference:



The availability of the **Model Views** toolbar options depends on the type of object selected. The options are, from left to right:

Icon	Usage	Shortcut	See Also
	Displays the appropriate Properties dialog for the item selected	<ul style="list-style-type: none"> • Double-click on the item • Enter 	
	Locates the selected object in any diagrams in which it has been used in the model, and either: <ul style="list-style-type: none"> • Displays the single diagram with the object highlighted or • Lists the several diagrams in which the object has been located 		
	Locates and highlights the selected object in the Project Browser		
	Creates a new <i>Model View</i> root node, and displays the New Model View dialog in which you enter the root node name		
	Creates a new <i>Views</i> folder in the currently-selected root node		
	Creates a new <i>Favorites</i> folder in the currently-selected <i>Views</i> folder		
	Creates a new Slideshow folder in the currently-selected <i>Views</i> folder		Diagram Slideshow ^[47]
	Creates a new <i>View</i> in the currently-selected <i>Views</i> folder, and displays the Create New View dialog to define the search that populates the View		Model Views Operations ^[46]
	Refreshes the selected <i>Model Views</i> root node, folder, <i>View</i> or <i>Favorites</i> ; for a <i>View</i> , this runs the Model Search defined in the View properties		Model Views Operations ^[46]
	Moves the currently-selected object up or down within its type ; you cannot move - for example - a package below a diagram, or a <i>View</i> above a <i>Favorites</i> folder		
	Displays a prompt to confirm deletion of the selected object and - if appropriate - its contents You cannot delete the original <i>Model Views</i> , <i>My Views</i> or <i>Recent Discussions</i> root nodes, or any technology-defined <i>Views</i>		
	Displays Help on Model Views		

4.1.4.2 Model Views Context Menus

Access: View | Model Views: Right-click on object

Reference:

The **Model Views** window context menus display different options, depending on which level of the **Model Views** hierarchy you right-click on. The options are described below:

Menu Option	Action	Shortcut	See also
Properties	<p>(Not the <i>My Views</i>, <i>Recent Discussions</i>, initial <i>Model Views</i> or <i>Technology-defined</i> root nodes)</p> <p>Display the appropriate Properties dialog for the selected object</p> <p>You can edit any of the properties, if required; changes to objects populated from the model are reflected in all other views (Properties window, diagrams, reports) of that object</p> <p>The Properties option for the <i>Recent Posts</i> folder displays the Recent Post Options dialog, which enables you to specify the number of days back from which to extract postings from the Team Review, and the number of postings to list</p> <p>The Properties option for a <i>Slideshow</i> folder displays the Slideshow Properties dialog, which enables you to automate the slide show and set the number of seconds for which each diagram is displayed</p> <p>If you deselect the Enable checkbox, you must press (Spacebar) to display each diagram</p>		Diagram Slide Show ^[47]
New Views Folder	<p>(Root node only)</p> <p>Display a prompt for the <i>Views folder</i> name and create the folder in the selected root node</p>		
Import Views From XML	<p>(Root node only)</p> <p>Prompt for the XML file location and create a new <i>Model Views</i> node to hold the imported Views</p>		Model Views Operations ^[46]
Export to XML (Views Only)	<p>(Root node only)</p> <p>Prompt for a file path and name, and copy all Views under the selected root node to an XML file at that location</p>		Model Views Operations ^[46]
Remove Model View	<p>(Not the <i>My Views</i>, initial <i>Model Views</i> or <i>Technology-defined</i> root nodes)</p> <p>Display a prompt to delete the selected user-defined Model View and, if confirmed, delete the root node and all contents</p>		
New Search Folder	<p>(View folder only)</p> <p>Display the Create New View dialog (similar to the</p>		Model Views Operations ^[46]

Menu Option	Action	Shortcut	See also
	View Properties dialog) for you to define the search that populates the View		
New Favorites Folder	(<i>View</i> folder only) Display the Create a new favorites based folder dialog, which prompts for the folder name		
New Slideshow	(<i>Model View</i> , <i>View</i> folder only) Display the Create a new slideshow dialog, in which you type the name of the slide show You must use the Properties dialog to define the properties of the slide show		Diagram Slide Show ^[47]
Open Search	(<i>View</i> only) Display the Model Search tab in the main work area, listing the full results of the search and giving access to all the facilities of the Model Search	Shift+Space	Customizing the Search View ^[48]
Refresh	Refresh the search and open the View or Postings Repository to show the elements or Posts retrieved by the search	Space	
Double Click Opens Search	Enable you to perform the Open Search function by double-clicking on the View If you deselect this option, double-click refreshes the search and opens the View to show the elements retrieved by the search		
Find Post	Open the Project Team Review and highlight the selected post		
Open Team Review	Open the Project Team Review at the top level (Category)		
Edit Connections	Display the Team Review Server Connections dialog, to select which review to open		Team Review Connections ^[23]
Remove Folder	(<i>View</i> folder only) Display a prompt to delete the selected Views folder and, if confirmed, delete the folder and all contents		
Remove View	Display a prompt to delete the selected View and, if confirmed, delete the View and all contents		
Remove Favorites	Display a prompt to delete the selected <i>Favorites</i> folder and, if confirmed, delete the folder and all contents		
Remove Slideshow	Display a prompt to confirm deletion of the selected slide show and, if confirmed, delete the slide show and all its diagrams		
Run Slideshow	Run the slide show in the Diagram View		Diagram Slide Show ^[47]
Run Slideshow Full Screen	Run the slide show in full screen mode, so that the slide show fills the whole screen		Diagram Slide Show ^[47]

Menu Option	Action	Shortcut	See also
Stop Slideshow	Cancel execution of the slide show running in the Diagram View	Esc	Diagram Slide Show ^[47]
In Project Browser	(Element / Diagram / Package object only) Highlight the selected item in the Project Browser		
In Diagrams	(Element / child Package object only) Locate the selected object in any diagrams in which it has been used in the model, and either: <ul style="list-style-type: none"> • Display the single diagram with the object highlighted or • List the several diagrams in which the object has been located 		
Remove Linked Item	(Element / Diagram / Package object only) Display a prompt to delete the selected object and, if confirmed, remove the object from the folder This has no effect on the object in the Project Browser or any diagrams You would not delete an object in a View, as it is replaced the next time the View is refreshed.		
Help	Display Help on Model Views		

4.1.4.3 Model Views Operations

This topic describes the operations you can perform in the Model Views window.

Access: [View | Model Views](#)

Topics:

Topic	Detail	See Also
Define View Search	<p>When you:</p> <ul style="list-style-type: none"> • Create a View, the Create New View dialog displays • Display the View properties, the View Properties dialog displays <p>These dialogs are identical</p> <p>In the Name field, type a name for the View</p> <p>In the Search field, either:</p> <ul style="list-style-type: none"> • Click on the drop-down arrow and select an existing search from the lists, or • Click on the (...) (Browse) button to display the Manage Searches dialog, edit an existing search or define a new one, then Close the dialog and select that search name in the Search field <p>A custom SQL search statement should return the <i>guid</i> and <i>type</i> of the object found so that Enterprise Architect can search for the selected</p>	Create & Modify Searches ^[48]

Topic	Detail	See Also
	<p>item in the Project Browser; for example:</p> <pre>SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE, Name FROM t_object</pre> <p>If required, in the Search Term field type a specific value to search for</p> <p>If you are working with the Corporate or extended versions of Enterprise Architect, and you want the search to run automatically and refresh the results, select the Refresh this search checkbox.</p> <p>The Frequency field has three sections, for hours, minutes and seconds; click on the appropriate section and use the up and down arrows at the end of the field to set the interval for refreshing the search results</p> <p>You can also set the refresh to display a pop-up notification if the search results change. To do this, select the Notify me when new results found checkbox</p> <p>Click on the OK button; the View is created (or updated) in a collapsed state</p> <p>When you expand the View, the search executes and populates the View</p>	
Display Recent Postings	<p>If a new post is added to the Team Review, and that post refers to any Model View items, that post is also added automatically to the <i>Recent Discussions</i> <Connection> <i>Recent Posts</i> folder in the Model View</p> <p>To open a post, double-click on the entry; the Team Review tab displays, showing the selected message</p> <p>You can control how many posts are displayed, and for what period of time, using the Model Views context menu for the <i>Recent Discussions</i> folder</p>	<p>Model Views 464</p> <p>Model Views Context Menus 467</p>
Move Objects Into Favorites	<p>Drag any required package, diagram or element from the Project Browser into the required <i>Favorites</i> folder</p>	
Move Objects Between Views	<p>Views and <i>Favorites</i> folders are fixed in the <i>Views</i> folder in which you create them, and you cannot move them</p> <p>However, you can:</p> <ul style="list-style-type: none"> • Copy (by dragging) objects from any View into any <i>Favorites</i> folder, and • Move (by dragging) objects between any two <i>Favorites</i> folders 	
Use Objects From Model Views	<p>To make use of an element, diagram or package held in any View or <i>Favorites</i> folder, click on the item and drag it into a diagram or a Team Review posting</p> <p>The item behaves in the same way as if you dragged it from the Project Browser.</p>	<p>Team Review 216</p>
Export/Import Views	<p>You export Views to create an XML file that you can:</p> <ul style="list-style-type: none"> • Import into another model as a user-created Model View or • Call from an MDG Technology Selection (MTS) file to access the Technology-defined View provided by the active MDG Technology. <p>The export and import functions are available from the Model Views root-node context menus</p>	<p>Model Views Context Menus 467</p>

Topic	Detail	See Also
	<p>When you use the export function, it acts on the complete set of View folders in the selected <i>My Views</i> root node, <i>Model Views</i> root node, or user-generated root node; you cannot export individual Views, nor can you export Favorites folders</p> <p>The function displays the Save As dialog, on which you browse for the directory location for the exported XML file, and specify the file name</p> <p>When you use the import function, it displays the Select Import Filename dialog on which you browse for the directory and XML file you want to import; the import creates a new Model View folder with the same name as the copied root node</p>	
<p>Set Up a Technology-Defined View</p>	<p>To set up the Technology-defined View for an MDG Technology, you:</p> <ol style="list-style-type: none"> 1. Create a user-generated Model View in Enterprise Architect while using the technology 2. Populate it with the required View folders and Views 3. Export the Views from that Model View as an XML file to an appropriate location 4. Create a call to the file from the technology's MTS file <p>Thereafter, any model for which the MDG Technology is active automatically displays those Views in a Technology-defined View</p>	<p>Working with MTS Files <small>[1073]</small></p>

4.1.4.4 Diagram Slide Show

The Slide show facility in the Model Views window enables you to create and run screen-based presentations of sets of diagrams:

When you run a slide show, it displays the diagrams in the folder in the sequence in which they are listed in the folder. After each diagram has been displayed, it is closed.

Access: [View | Model Views](#)

Use to:

- Create a slide show as a folder within a Views folder under a Model View node
- Run a slide show manually
- Run a slide show automatically with each diagram displaying for a period that you define
- Display a slide show within the Enterprise Architect work area
- Display a slide show in full screen mode
- Delete a slide show

How to:

To create a slide show, follow the steps below:

Step	Action	See also
1	Under a Model Views node, click on the Views folder to contain the slide show	
2	Either: <ul style="list-style-type: none"> • Right-click on the Views folder and select the New Slideshow option 	

Step	Action	See also
	<p>or</p> <ul style="list-style-type: none"> Click on the New Slideshow folder icon in the Model Views toolbar <p>The Create a New Slideshow Folder dialog displays</p>	
3	In the Name field, type the name of the slide show	
4	<p>Click on the OK button</p> <p>The new slide show folder is added to the selected View folder</p>	
5	Dock the Project Browser window separately from the Model Views window.	Dock Windows ^[103]
6	<p>Drag the diagrams to be displayed as part of the slide show from the Project Browser into the new slide show folder</p> <p>Organize the diagrams in the order in which they are to be displayed, using the up/down green arrows in the Model Views toolbar</p> <p>To remove a diagram that is not required, right-click on it in the slide show folder and select the Remove Linked Item context menu option</p> <p>The diagram items in the slide show folder are links to the diagrams in the Project Browser, not copies of the diagrams or the diagrams themselves; any actions you take in the slide show have no impact on the original diagrams</p>	Model Views Toolbar ^[466]
7	Close the folder	

To define how the slide show is to operate, automated or manual, follow the steps below

Step	Action	See also
1	<p>Right-click on the slide show folder and select the Properties context menu option</p> <p>The Slideshow Properties dialog displays</p>	
2	<p>To automate the slide show, select the Enable checkbox</p> <p>To run the slide show manually, deselect the checkbox</p>	
3	If you are automating the slide show, in the Time field type the number of seconds that each diagram is to remain on display	
4	Click on the OK button	

To run a slide show of diagrams, either in the Diagram View or full screen, follow the steps below:

Step	Action	See also
1	<p>Right-click on the slide show folder and select either the:</p> <ul style="list-style-type: none"> Run Slideshow option, to run the slideshow in the Diagram View Run Slideshow Fullscreen option to run the slideshow using the full screen <p>The first diagram in the slide show displays</p>	

Step	Action	See also
2	If you have set up the slide show to run automatically, you can leave it to display the diagrams as defined You can also moderate the slide show using manual commands	
3	If you are controlling the slide show manually, right-click on a slide The following toolbar displays: 	
4	Control the slide show using the toolbar icons and other aids, as follows: <ul style="list-style-type: none"> • Display the next slide - , (Spacebar) or (↑) • Display the previous slide - , (←) • Display the first slide - , (↑) • Display the final slide - , (↓) • Pause the slide show -  • Resume the slide show -  • Stop the slide show -  or, in Diagram View, right-click on the slide show folder and select the Stop Slideshow option 	

To delete a slide show, follow the steps below:

Step	Action	See also
1	Right-click on the slide show folder and select the Remove Slideshow context menu option Enterprise Architect prompts you to confirm the deletion	
2	Click on the Yes button The folder and its list of links to diagrams in the model is removed	

4.1.5 The Pan & Zoom Window

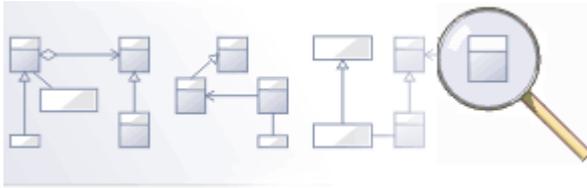
The Pan & Zoom window provides a 'birds-eye' view of diagrams. It enables you to navigate quickly around large diagrams.

Access: **View | Pan & Zoom**

Reference:

Topic	Detail	See also
View	The window shows a reduced image of the whole diagram The shaded box represents the viewed area, displayed on the open diagram	
Toolbar	The toolbar provides the following functions (in order): <ul style="list-style-type: none">• Zoom In• Zoom Out• Zoom to fit diagram• Zoom to fit page• Zoom to 100%• Zoom Slider	
Use	Open the required diagram Move the cursor onto the shaded box inside the Pan & Zoom window Hold down the mouse button as you move the cursor to move the shaded box over the open diagram To zoom, use either the Zoom Slider or the buttons located on the tool bar	

4.2 Search: Finding Information



Searching involves using ad hoc search terms and various search types to locate elements and information within your model, regardless of location or context.

The result sets can be used to manage elements, to create documentation or to find the location of elements within the complete model structure.

Topics:

Topic	Link
Model Search - how to use the searching capability within Enterprise Architect to find and display various kinds of information	Model Search ^[475] Model Search Context Menu ^[477] Model Search Toolbar ^[478] Pre-defined Searches ^[479] Customizing the View ^[480]
Create and Modify Searches - how to modify and work with search definitions	Create & Modify Searches ^[482]
Adding Filters - further modifications to searches	Add Filters ^[487]
Diagram Filters- how to suppress or highlight information in a diagram using customizable filtering mechanisms based on element properties such as status and complexity	Diagram Filters ^[489]

4.2.1 Model Search

The Model Search allows you to quickly navigate and find Elements within your model. It lists each Element that meets the search criteria you specify within the search terms and search type. Many different kinds of searches are already built into Enterprise Architect, and it is possible to create very detailed and specific searches of your own. It is even possible to export and share your favorite searches.

Access: **Edit | Model Search** (**Ctrl + Alt + A**) or (**Ctrl + F**)

Use to:

- Search for elements within your model based on various criteria
- Generate targeted reports based on your result set
- Manage result sets - for example set properties and delete elements

Field	Usage	See also
-------	-------	----------

Search Term	The word, phrase or characters to search on	
Search	<p>The Model Search window provides a built in set of searches for your convenience; the Search drop-down list provides several pre-defined searches including.</p> <ul style="list-style-type: none"> • Simple • Extended • Find Orphans • Recently Modified Elements <p>Search filters enable you to perform customized searches on a Search Term in order to locate model elements having specific characteristics</p>	Pre-defined searches ^[479]
Run	Run the search using the entered text and selected Search	
Options	<p>The Options button displays the Search Options submenu, which enables you to display the search results as a tab of the Output window rather than in the Model Search View</p> <p>An advantage of moving the search results to the Output window is that you can select items from the search results and drag them onto a diagram, which you cannot do when the results are in the Model Search View</p> <p>If you select the Dock in Output Bar menu option, when you next display the menu this option becomes Dock in Main View</p> <p>The Search Options submenu also provides the means of performing advanced searches on your project, and displaying project metrics</p>  <p>The Options button also enables you to perform advanced searches</p>	The System Output Window ^[127] Advanced Search Options ^[486] Estimating Project Size ^[404]
	Show the Search Builder facility	Create Search Definitions ^[484]
Result List	Displays the results of your search	

Learn More:

- Display the properties of an Element (**Double Click or Enter**)
- Drag an item(s) onto a diagram (**Ctrl + drag a selected item**)
- Toggle between the Notes and the **Model Search** window (**Ctrl + Shift + Enter**)
- Open the Linked Document window for the selected Element (**Ctrl + Alt + D**)
- Select All Items (**Ctrl + Alt + A**)

Notes:

You can access the **Model Search** facilities and perform specific searches indirectly, from Add-Ins, MDG Technologies, from a hyperlink and from a shortcut to access your model. This entails setting up a search profile either in the appropriate tool, or as an XML file accessed by the tool.

Searching from	See
An MDG Technology Selection (MT) File (using an exported search definition)	Working With MTS Files ^[1073] Create & Modify Searches ^[482]
A Login Shortcut	Project Shortcuts ^[140]
An Add-In	Add-In Search ^[1974]
A Hyperlink	Hyperlinks ^[1292]

Running a custom or diagram based search disables some context menu options.

4.2.1.1 Model Search Context Menu

You can select elements or diagrams in the Model Search and perform various operations on them, as well as simply dragging the item into a Team Review post.

Access: Results pane, right-click on an element

Use to:

- Generate reports from search results
- Print search results
- Copy and work with search results
- Manage elements returned by a search

Reference:

Menu Option	Action	Shortcut	See also
Properties	Display the Properties dialog for the element		
Copy Selected to Clipboard	Copy the selected item to the MS Windows clipboard so that it can be pasted to a document, spreadsheet or email		
Documentation	<p>Generate an RTF report; you have two options:</p> <ul style="list-style-type: none"> • Generate a separate report on each selected object in the Model Search • Generate one report on all selected objects <p>In either case, the Generate RTF Documentation dialog displays</p> <p>If you generate the report using a custom SQL search, the SQL must include <i>ea_guid</i></p>		<p>Generate RTF Documentation Dialog ^[1732]</p> <p>Create Search Definitions ^[484]</p>

	AS <i>CLASSGUID</i> and the object type		
Create Linked Document (Edit Linked Document)	Create (or edit) a linked document (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions)	Ctrl + Alt + D	Link Document to UML Element ^[730] Edit Linked Documents ^[731] Linked Documents ^[729]
Delete Linked Document	Delete an existing linked document (Only displays if the element has a linked document)		
Print	Print out the filtered results		
Clear Results	Clear the search results from the Model Search		
Find in Diagrams	Display the diagram that uses the element or, if the element is used in multiple diagrams, display a list of diagrams to choose from		
Find in Project Browser	Highlight the element in the Project Browser		
Bookmark Selected	Bookmark the element		
Delete Selected	Delete the selected element from the Model Search		
Close	Close the Model Search		
Help	Display this Help topic on the Model Search		

Learn More:

- [Add a New Post](#)^[222]

4.2.1.2 Model Search Toolbar

The Model Search toolbar displays at the top of the Model Search window, enabling you to operate on the results of your model search.

Access: **Edit | Model Search**

Use to:

- Edit the text of notes for an item
- Delete selected items from the model
- Print the search results
- Generate and print an RTF report on selected items
- Hide or abbreviate notes text in the output
- Hide or show the filter bar under the column headings
- Display Help text

Reference:



Action	Usage	Shortcut	See also
Edit Notes	For the selected item, opens the Notes window (if it is not already open) so that you can edit the text of the notes	Ctrl + Spacebar	Notes ^[769]
Delete	For a selected item or group of items, deletes them from the model; refresh the project to check that the items have been deleted	Ctrl + D	
Print	Prints the complete set of search results		
Rich Text Report	For a selected item or group of items, generates and prints an RTF report		Generate RTF Documentation Dialog ^[1732]
View Notes	Displays a short menu that enables you to select whether, for all items, to: <ul style="list-style-type: none"> • Hide any Notes text from display in the search results • Display the first few words of the Notes text in the search results • Display the full Notes text in the search results 		
Toggle Filter Bar	Hides or reveals the filter bar underneath the column headings		Customizing the Search View ^[480]
Help	Displays the Enterprise Architect Help, starting with the Model Search Help topic		

4.2.1.3 Pre-defined Searches

The following pre-defined searches are provided with Enterprise Architect, and are listed in the *Built-In* category in the **Search** drop-down list.

Search	Description	See also
Simple	Searches the Name , Alias and Notes fields of all elements for the given search term	
Extended	Searches many additional fields relating to the element, including Attributes , Operations , Tagged Values and Test Cases	
Element Name	Searches for an exact match against the element name, alias, method or operation; the default search in the Search in Model menu option in the Code Editor	Code Editor Context Menu ^[1404]

Attribute Details	Searches for elements with attributes relating to the search term, including Tagged Values, constraints, and common attribute data fields	
Find Orphans	Searches for orphaned elements throughout the model, with the ability to filter on common element fields using a search term An 'orphaned' element is an element that does not appear on any diagram in the model	
Failed Internal Tests	Searches for elements containing internal test cases where the search term is in any common Test Case field and the Status value is <i>Fail</i>	
Method Details	Searches for elements with operations and methods relating to the search term, including Tagged Values, constraints and common operation and method data fields	
Responsibility	Searches for elements with internal responsibilities/ requirements where the search term relates to any common Responsibility/Requirement field	
Resources	Searches for elements with assigned resources where the search term relates to any common Resource field	
Requirements	Searches for Requirement element types where the search term relates to any common element field	
Find Bookmarked Elements	Searches for elements that have been bookmarked, anywhere in the project	
Recently Modified Elements	Searches for elements that have been recently modified, anywhere in the project The search term relates to any common element field The default is to show elements modified in the last three days, but you can set an alternative interval by typing the appropriate number of days in the Search Term field	
Recently Modified Diagrams	Searches for diagrams that have been recently modified, anywhere in the project The search term relates to any common diagram properties field The default is to show diagrams modified in the last three days, but you can set an alternative interval by typing the appropriate number of days in the Search Term field	
My Checked Out Packages	Searches for packages that are marked as checked out by the currently-logged in user	

4.2.1.4 Customizing the Search View

This topic explains how to customize the display of search results after they are generated.

Topic	Detail	See also
Sort and Select	In the Model Search you can: <ul style="list-style-type: none"> Sort the items by any column value in ascending or descending order, by 	

Topic	Detail	See also
	<p>clicking on the column header</p> <ul style="list-style-type: none"> • Display element or diagram properties, by double-clicking on the item • Select: <ul style="list-style-type: none"> • An element or diagram by clicking on it • Several individual elements or diagrams by holding (Ctrl) as you click on them • A range of elements or diagrams by holding (Shift) as you click on the first and last in the range • All elements or diagrams in the list by pressing (Ctrl+A) 	
<p>The View Header</p>	 <p>The View Header defines the columns of information that are presented by the Model Search, and the order in which data items are presented</p> <p>To review the column headings:</p> <ul style="list-style-type: none"> • Right-click on the header to display the Field Chooser context menu option; the Field Chooser dialog displays <p>This enables you to add columns to or remove columns from the output; between them, the View Header and Field Chooser dialog show the full range of column headers available</p> <p>To add a column heading to the View Header:</p> <ul style="list-style-type: none"> • Drag it from the Field Chooser dialog onto the header to the position you want the column of data to display • When you have selected the column headings you require, click on the red cross in the top right corner of the Field Chooser dialog to close it <p>To remove a column from the output:</p> <ul style="list-style-type: none"> • Drag the column heading downwards out of the View Header <p>To change the sequence of columns:</p> <p>Drag the required column header left or right, as appropriate</p>	
<p>Value Grouping</p>	<p>You can organize the reported data hierarchically according to the value of one or more of the column categories</p> <p>You might organize the data by Type, and within each Type by Name; if you then click on any of the other column headings, the data within this grouping is further sorted with the values of the selected column (for example, Created) in ascending or descending order</p> <p>To set up the value grouping, drag the column heading representing the primary grouping (such as Author) onto the Drag a column header here to group by that column field</p> <p>Now drag the column heading for the next level of grouping (such as Status) to the right of the first heading; the two heading titles display as connected blocks, as shown below:</p> 	

Topic	Detail	See also
	<p>You can, if required, add further levels of grouping by dragging other column headings onto the hierarchy (such as Created), and restructure the order by dragging existing or additional headings into the level you want them to hold</p> <p>For example, you could make Type the secondary grouping by dragging it to the right of Author, or drop Status between Author and Type</p> <p>To remove a grouping level, drag the appropriate column heading out of the sequence and below the View header; any subordinate groupings move up a level</p>	
Filter Bar	<p>The Filter Bar displays underneath the column headings; you can hide or show it using the Toggle Filter Bar icon in the toolbar.</p> <p>The Filter Bar consists of a row of fields, one field per column; as you type a value into one of these fields, the Model Search results are automatically filtered to show entries with a value in the corresponding column that match the value in the field. The filter is case sensitive.</p> <p>For example, if you type the letter F in the Filter Bar field under the Author column, the results immediately reduce to entries for which the author name includes names such as Fred, Frank, Felix, Felicity, Fairmile, Farmer or Fitzgibbon (but none where the name includes f only as a lower case letter, such as Rufus)</p> <p>If you then go on to type Fi, the results immediately reduce again to only entries for which the author name includes names such as Fitzgibbon.</p>	

4.2.1.5 Create & Modify Searches

You provide search filters and create new search definitions using the Search Builder panel, displayed underneath the **Search Term** and **Search** fields on the Model Search window.

Access: **Edit | Model Search: Show Search Builder** ()

Use to:

- Create new search definitions and edit them

Search Builder Toolbar:

The search builder toolbar enables you to configure the system-provided searches, and to create and edit your own searches

The toolbar icons, from left to right, provide access to the following functions:

Option	Usage	See also
New Search	Create a new search definition, with new search criteria	Create Search Definitions ^[484]
Save Search	Save a modified or new search	
Copy	Copy the existing search selected in the Search field, to modify	

Option	Usage	See also
Search	You can copy: <ul style="list-style-type: none"> • Searches you or other users have created • Predefined searches and edit the copies, but you cannot edit the predefined searches themselves 	
Restore Default	Restore any changed parameters to the default settings and format	
Delete Search	Delete the search definition from the Search drop-down list	
Add Filter	Add a new set of parameters to filter the search on	Add Filters ^[487]
Remove Filter	Delete the selected filter from the search	
Export Search	Display a selection box that enables you to select searches to export to an external directory as an XML Search file	
Import Search	Display the Windows Directory Explorer Open dialog to enable you to import searches as XML Search files from an external directory	
Search Options	Display the Advanced Options dialog, to define where the search should operate and how the search should match filters This icon is available only for searches created with the Query-Builder, and not for those created with the SQL Editor	Advanced Search Options ^[486] Create Search Definitions ^[484]

Search Builder:

You use the main body of the Search Builder panel to configure the element search filters that are contained in the selected search.

The Search Builder panel defaults to a **Simple** search, which changes as you select to create a new search or select the search to modify.

Column	Usage	See also
Search In	Select the type and name of each element feature (the element field name) to search on.	
Condition	Select the condition of the search parameter (the condition placed on the field value) The available options are Contains, Equal To, Not Equals and One Of	Fields and Conditions ^[489]
Look for	Specify the search term (the actual value or delimiting value) to perform the conditional search on This value can pertain to the selected element type; for example, the value could be a date for <i>DateCreated</i> or a text value for other fields The search term can contain multiple values, separated by commas	Fields and Conditions ^[489]
Required	Indicate whether the field is mandatory - that is, whether the search results must include elements with your search term in that field The following provides example of using the Required checkbox, illustrated	

Column	Usage	See also
	<p>by the default Simple search, which checks the Name, Alias and Notes fields of all elements:</p> <ul style="list-style-type: none"> • If the Required checkbox is not ticked for any field, then if the search term is found in the Name, Alias OR Notes field for an element, that element is listed in the results • If the Required checkbox is ticked for all fields in the search, then the search term must be found in all of those fields before the element is listed in the results; that is: Name AND Alias AND Notes • If the Required checkbox is ticked for some fields but not others, the search term must be found in all fields for which the checkbox is ticked, and <i>at least one</i> of the fields in which the checkbox is not ticked; that is: Name AND (Alias OR Notes) 	

4.2.1.5.1 Create Search Definitions

User-defined searches are created using the Search Builder panel, and are stored in the Program Files directory (not in the project repository).

Access: [Edit](#) | [Model Search: Options](#) | [Manage Searches](#)

Use to:

- Create a search definition using the SQL Editor, Query Builder, or an Add-In Search

How to:

To create a new search definition, follow the steps below:

Step	Action	See also
1	Click on the New Search icon in the toolbar The Create New Search Query dialog displays	
2	In the Search Name field, type a name for the search	
3	Select the radio button for the type of search you require: <ul style="list-style-type: none"> • Query Builder - provides an interface that enables you to design your own search • SQL Editor - enables advanced users to directly write SQL Select statements • Add-In Search - enables you to supply the name of your Add-In and a method (for example <i>MyAddin.RunThisMethod</i>), which is called whenever the search is run; this search can be exported and distributed as a part of your Add-In 	Enterprise Architect Add-In Model ^[1967]
4	Click on the OK button, and refer to the following table	

Search Types:

Search	Description	See also
<p>Query Builder</p>	<p>Your search definition appears as being selected in the Search drop-down field</p> <p>You can now click on the Add Filter toolbar icon to add filters</p>	<p>Add Filters [487]</p>
<p>SQL Editor</p>	<p>The Custom SQL dialog displays, enabling you to input your <i>SELECT</i> statement</p> <p>The SQL editor is based on the common Code Editor, and provides an intellisense autocompletion list populated with Enterprise Architect's repository structure; to display the autocompletion list, press (Ctrl+Spacebar)</p> <p>Enterprise Architect also enables you to use #xxx# macros as string replacers in <i>WHERE</i> statements, so that the same search can be used by different people in different environments; these macros include:</p> <ul style="list-style-type: none"> • #WC# - Gets the appropriate wild card for the current database, and so enables the search to be performed on models on different databases; for example: <i>t_object.Name LIKE '#WC#Test#WC#'</i> • #Author# - Takes the value of the Author field in the Options dialog General page, and enables the defined search to be performed on objects created by that user (this value can be manually re-set in the Options dialog) • #DB=<DBNAME># where <DBNAME> can be one of the following: <ul style="list-style-type: none"> • MYSQL • JET • ORACLE • SQLSVR • ASA • OPENEDGE • POSTGRES <p>For example: <i>#DB=ORACLE# t_object.ModifiedDate >= (SYSDATE - INTERVAL '<Search Term>' DAY)</i></p> • #UserName# - Gets the name of the person logged into version control; for example: <i>t_package.PackageFlags LIKE '#WC#VCCFG=#WC#CheckedOutTo=#UserName##WC#'</i> (this is from Enterprise Architect's built in search <i>My Checked Out Packages</i>) • #Package# - gets the currently-selected package's package_ID • #Branch# - gets the IDs of the child packages of the currently-selected package <p>For all Enterprise Architect functions in which you use a custom SQL statement (such as RTF reporting or Model Views) the statement must return the <i>guid</i> and</p>	<p>Code Editors [1394]</p>

	<p>type of the object found so that Enterprise Architect can search for the selected item in the Project Browser; for example:</p> <pre>SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE, Name FROM t_object</pre> <p>You can extend the usability of your SQL searches using the aliases <i>CLASSGUID</i> and <i>CLASSTYPE</i>, which enable Enterprise Architect to display the Properties dialog and icon for elements, connectors, attributes or operations, as well as selecting them in the Project Browser</p> <p>Some simple examples for using these aliased fields are provided below:</p> <ul style="list-style-type: none"> • <code>SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE, Name FROM t_object</code> • <code>SELECT ea_guid AS CLASSGUID, Connector_Type AS CLASSTYPE, Name FROM t_connector</code> • <code>SELECT ea_guid AS CLASSGUID, 'Operation' AS CLASSTYPE, Name FROM t_operation</code> • <code>SELECT ea_guid AS CLASSGUID, 'Attribute' AS CLASSTYPE, Name FROM t_attribute</code> <p>When you have defined the <i>SELECT</i> statement, click on the Save button to save this search; the search is then available from the Search drop-down list</p>	
Add-In Search	<p>Type in:</p> <ul style="list-style-type: none"> • The name of your Add-In • A period (full stop) and • The name of the method to be called (for example, <i>MyAddin.RunThisMethod</i>) <p>Your search is automatically saved and available from the Search drop-down list</p>	

Learn More:

- [Customizing the Search View](#)^[480]
- [Create & Modify Searches](#)^[482]

4.2.1.5.2 Advanced Search Options

Access: **Edit | Model Search: Options | Manage Searches: Search Options (toolbar icon)**

Use to:

- Define how the customized search should operate

Reference:

Action	Usage	See also
Return matching items	To run the search across the entire model (the default)	

Action	Usage	See also
for the entire model		
Return matching items for the selected Package	<p>To run the search on a specific package, selected from the Project Browser</p> <p>Navigating the Project Browser does not change your search results until you click on the Run button; that is, to search different areas of the project:</p> <ul style="list-style-type: none"> • Click on the first required package in the Project Browser • Click on the Run button • Check the result • Click on another package in the Project Browser • Click on the Run button again 	
All Tables	To ensure that the search only retrieves elements that match every check in the search	
At Least One Table	To ensure that the search retrieves elements that match at least one of the checks in the search (the default)	

4.2.1.5.3 Add Filters

Access: [Project | Documentation | Rich Text Format \(RTF\) Report: Add Filter Edit | Model Search: Options | Manage Searches: Add Filters \(toolbar icon\)](#)

Use to:

- Add filter criteria to a model search or an RTF report template

Reference:

Option	Action	See also
Search On	<p>Select items to build up search filters on any information about an object</p> <p>The following is a list of what is available, before you have defined a search</p>	

Option	Action	See also
	<div data-bbox="438 338 869 936" style="border: 1px solid black; padding: 5px;"> <p>Element Diagram Attribute Attribute.AttConstraint Attribute.AttTagValue Change Custom Property Constraint Method Method.MethodTagValue Method.Parameter Method.PostCondition Method.PreCondition Method.Parameter.ParamTagValue File Issue Defect Scenario TagValue Task Test Responsibility Resource</p> </div> <p>If you are adding filters to an existing search, the list contains only items appropriate to the initial filter</p> <p>For example, if the initial filter is on diagram properties, the list for any subsequent filters for the search only contains the Diagram option</p>	
Include	Select each field item to include in your search (select the checkbox)	
Field	Identify the name of the field to search The list presents items specific to the filter <i>Search On</i> item	Fields and Conditions ^[489]
Condition	Specify the condition of the search parameter	Fields and Conditions ^[489]
Value	Type a value pertaining to the selected element field For example, the value could be a date for <i>DateCreated</i> or a text value for other fields The search term can contain multiple values separated by commas	Fields and Conditions ^[489]
Required	Select a particular field to generate a result set that <i>must</i> contain your search term in that field	
Check All	Select all the items to include them in the search definition	
Uncheck All	Deselect all the items to omit them from the search definition	
OK	Apply the filter. The fields selected are added to the search definition	

Notes:

- You can add multiple search definitions as necessary; however, if you select the **Required** field in multiple definitions the search rapidly becomes impractical
- Multiple search definitions are better for 'and/or' searches

Learn More:

- [Create Search Definitions](#)^[484]
- [Create & Modify Searches](#)^[482]
- [Generate RTF Documentation Dialog](#)^[1732]

4.2.1.5.3.1 Fields and Conditions

When you click on a condition for a particular field, a selection of condition options becomes available. The list of options varies dependent on the type of information required.

For some conditions, the **Value** field contains an ellipsis (...). Click on this to display a selection dialog. This dialog varies dependant on the information required.

Learn More:

- [Create Search Definitions](#)^[484]
- [Add Filters](#)^[487]

4.2.2 Diagram Filters

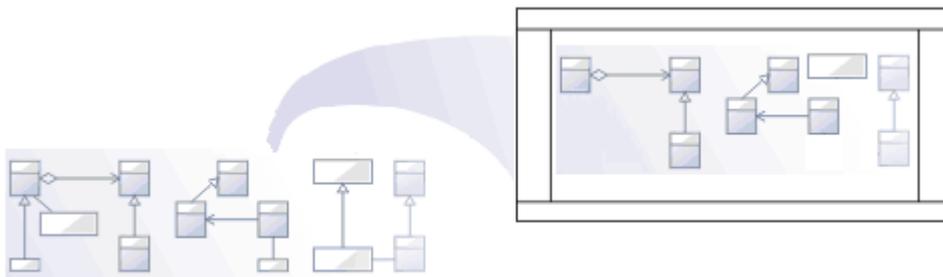


Diagram Filters (Dynamic Visual filters) enable you to modify the display of diagram components, so that the relevant items are immediately identified for the reader's attention without damaging the structure and integrity of the model. Currently the feature operates on elements and connectors, and filters according to properties such as Author, Status, Date Created or Stereotype.

Access: [View | Diagram Filters](#)

Topics:

Topic	Detail	See also
Use	<p>Use Diagram Filters to tailor the display of diagrams:</p> <ul style="list-style-type: none"> • For different users, so that - for example - technical staff and stakeholders each have a view that highlights the information pertinent to them • To show what elements have been recently developed or changed • To show which part of a model was developed by a particular person • To show which parts of a diagram are at a particular phase, status or version <p>Filters for elements can also be applied to the Package Browser and Diagram List; the <i>Select</i> filter effect does not, however, make a difference in</p>	<p>Package Browser^[456]</p> <p>Diagram List^[462]</p> <p>Work With Diagram Filters^[497]</p>

Topic	Detail	See also
	these contexts	
Develop Diagram Filters	<p>You create and define as many filters as you require, setting up each filter by defining which element or connector properties to specifically check for and (depending on how you set up the filter parameters) whether to include or exclude elements or connectors having particular property values</p> <p>You cannot combine element and connector properties in one filter, but you can apply an element filter and a connector filter to a diagram at the same time</p> <p>You can select to:</p> <ul style="list-style-type: none"> • Mute the irrelevant items in grayscale or a faded color • Hide the irrelevant items completely, or • Highlight (with a hashed line) the items that are relevant 	Work With Diagram Filters 49↑
Filter Action - Mute or Hide	<p>If you select to Mute or Hide items, the action of the filters is to <i>exclude</i> items that <i>do not</i> match the parameters rather than <i>include</i> items that do</p> <p>If, for example, you selected to filter on element name, looking for elements with the word <i>Class</i> in the name, the filter would apply the following logic:</p> <p style="padding-left: 40px;">Does Name contain string <i>Class</i>? If No, apply effect; if Yes, take <i>no</i> action</p> <p>The elements you want are therefore what is left on the diagram, rather than what was operated on</p> <p>The filter effect remains in force while you do other work on the diagram, until such time as you disable the filter</p>	Work With Diagram Filters 49↑
Filter Action - Highlight	<p>If you select the <i>Highlight</i> effect, the logic is reversed:</p> <p style="padding-left: 40px;">Does Name contain string <i>Class</i>? If No, take <i>no</i> action. If Yes, apply effect.</p> <p>In this case, the filter effect is not permanent, and clicking off the items deselects them</p> <p>This effect is excellent for selecting items having the same characteristics across a large diagram, to be processed in a single task; instead of having to locate the items and select them with (Shift) +click individually, you can apply the filter</p> <p>If you inadvertently lose the selection by clicking off the items, you can get it back again almost immediately by re-applying the filter.</p>	Work With Diagram Filters 49↑
Application	<p>You can use filters singly, in sequence, or in combination; for example, you could:</p> <ul style="list-style-type: none"> • Set up a filter for immediate use on a diagram, and modify that filter as you review the diagram so that you highlight elements with different values for the same properties - perhaps, by filtering on Phase, to compare 'as-is' and 'to-be' elements • Set up a filter and leave it active so that all diagrams you display are automatically filtered the same way • Set up a series of filters to use: <ul style="list-style-type: none"> • in one or more sequences to progressively highlight a diminishing set of items, or • alternately to highlight contrasting views of the diagram 	

Learn More:

- [A Quick-Start Guide to Using Diagram Filters](#)

4.2.2.1 Work With Diagram Filters

The Diagram Filters window is a dockable window, which you can move around or fix next to the Diagram View while you activate, deactivate and edit the filters.

Access: [View | Diagram Filters](#)

Use to:

- Create and edit filters
- Set the effect of the filter
- Enable and disable filters
- Delete filters

How to:

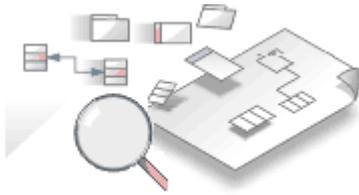
To create a filter to be applied to your diagrams, follow the steps below:

Step	Action	See also
1	Perform one of the following actions: <ul style="list-style-type: none"> • On the Diagram Filter toolbar, click on the New Filter icon - the first on the left • Right-click on the list panel and select the New Filter context menu option The Create New Diagram Filter dialog displays	
2	In the Enter Filter Name field, type a name for the filter, then click on the OK button The Diagram Filter dialog displays	
3	In the Search On field, either: <ul style="list-style-type: none"> • Use the default value of Element or • Click on the drop-down arrow and select Connector If you select Connector , the dialog changes to offer a smaller set of properties	
4	Scroll through the properties to filter on, and select the checkbox against each property you require	
5	For each property, click on the Condition field and select, from the drop-down list, the comparison condition to be applied Consider how the combination of Condition - Equal To / Not Equal To and Filter Effect might affect the results on the diagram	Diagram Filters ^[489]
6	For each property, double-click on the Value field and type or select any specific value to filter on	
7	Click on the OK button to save the filter and return to the Diagram Filters window	

Other Operations:

Topic	Detail	See also
Edit a Filter	<p>To edit an existing filter on the Diagram Filters window, either:</p> <ul style="list-style-type: none"> • Double-click on the filter name • Click on the filter name and select the Properties icon from the toolbar (the second icon from the left), or • Right-click on the name and select the Properties context menu option <p>The Diagram Filter dialog displays; adjust the filtered fields as described above</p>	
Change the Name of the Filter	<p>Right-click on the name and select the Change Name context menu option</p> <p>The Create New Diagram Filter dialog displays</p> <p>Type over the existing name with the new name, and click on the OK button</p>	
Set Effect of Filters	<p>To set how your filters identify selected items on your diagrams, click on the drop-down arrow of the toolbar Filter Effect field, and select one of the following options:</p> <ul style="list-style-type: none"> • Fade - display all items that do <i>not</i> match the filter criteria in a pale version of the diagram background color. • Gray Scale - display all items that do <i>not</i> match the filter criteria in pale gray. • Hide - conceal all items that do <i>not</i> match the filter criteria. • Select - select and highlight (with a hashed line) all those items that <i>do</i> match the filter criteria. 	
Enable Filters	<p>To enable a filter so that it takes immediate effect on your diagrams, select the check box against the filter name</p> <p>You can select more than one filter at a time, to combine their effects</p>	
Disable/ Clear Filters	<p>To disable a filter and clear the effect of the filter on the diagram so that it displays in full, clear the checkbox</p> <p>To disable all filters, either:</p> <ul style="list-style-type: none"> • Click on the Reload Filters icon in the toolbar (third option from the left), or • Right-click on the list panel and select the Reload Filters context menu option 	
Delete a Filter	<p>To remove a filter from the Diagram Filters window, either:</p> <ul style="list-style-type: none"> • Click on the filter name and click on the Delete icon in the toolbar, or • Right-click on the filter name and select the Delete Filter context menu option 	

4.3 Trace: Tracking Dependencies



Traceability helps to identify the way a given requirement or process has been implemented in a system, enabling you to follow the path of dependencies from the initial request, through a modeled solution and up to the deployed physical system or process.

Topics:

Topic	Link
<p>Application - A well developed model provides full traceability in any direction - providing answers to questions such as "why was that particular hardware used in the solution"; assuming there is traceability back to a requirement that specified particular response times or functional capability</p> <p>The result sets can be used to manage elements, to create documentation or to find the location of elements within the complete model structure</p>	
<p>Tools Overview - There are various tools in Enterprise Architect that enable you to trace the definition and implementation of a process from initial requirement to generated code or technical deployment, or vice versa</p> <p>A good starting point is to look at the topics listed to the right, which detail some of the more useful tools</p> <p>The Traceability window, in particular, is designed to provide very detailed information about an elements immediate and distant relationships and dependencies</p>	<p>The Traceability Window ^[495]</p> <p>Relationship Matrix ^[496]</p> <p>Gap Analysis Matrix ^[510]</p> <p>The Relationships Window ^[504]</p> <p>The Project Browser ^[441]</p> <p>Example Diagram ^[505]</p> <p>Traceability Tools ^[494]</p>
<p>Tracing Transformations - If you have performed any Transformations in developing your model and code, Enterprise Architect automatically creates <i>Transformation Dependency</i> connectors that you can trace - with the Traceability window - to establish what objects and code have been generated from each PSM element, or what the initial PSM element was for a generated object</p> <p>Whether you use transformations or develop the stages of the model in other ways, you can build up a range of <i>Traceability diagrams</i> (<i>Custom</i> diagrams) to identify the development pathway and the dependencies between entities such as Requirements, Use Cases, Classes, Packages, Test Cases and other model artefacts, or possibly between these entities and the overall business process model</p>	<p>Model Transformation ^[1301]</p> <p>The Traceability Window ^[495]</p> <p>Example (Traceability) Diagram ^[505]</p> <p>A Complete Business Process ^[1198]</p>

Learn More:

- [Packages and Elements](#) ^[506]

4.3.1 Traceability Tools

The model structure and a Traceability diagram act as starting points for tracing the definition, design and implementation of a feature of a system or process.

By applying tools such as the Relationship Matrix and Traceability window to these, you can follow threads throughout the model to determine how the feature is implemented and tested.

You can also obtain information on what elements realize and are realized by the elements in a given package, using the Dependency report and Implementation report, respectively.

Topic	Detail	See also
Traceability Window	<p>The Traceability window is a most useful and versatile traceability tool; starting with a Traceability diagram or a package structure in the Project Browser, you use the Traceability window to quickly explore the relationship chain of which any element is a component</p> <p>When you click on an element, it immediately becomes the top point in the Traceability window</p> <p>When you click on the background of a diagram, <i>all</i> elements in the diagram are listed in the Traceability window, and you can follow the threads starting at each element through the diagram</p> <p>If you require a rapid, broad-brush view of relationship flows in the project structure, starting with a general list of - say - all functional Requirements, you can use a combination of Model Search, Project Browser and Traceability window; this is a powerful tool for scanning your project, identifying how elements have been organized, and how they interact</p> <p>For example: the Model Search would list all the requirements; you could rapidly click on each element and immediately see in the Project Browser where it has been grouped, and at the same time - in the Traceability window - how that element interacts with other elements in the model</p> <p>By moving the cursor around a diagram or the Project Browser, and/or changing the relationship type combinations in the Traceability window, you can quickly see how elements are connected and how they influence each other</p> <p>For example, you can see that - say - REQ017 is realized by two Use Cases, so you might then explore what else influences and is influenced by these two Use Cases; the Traceability window takes you well beyond what is likely to be depicted on any single diagram.</p> <p>If you have used transformations to develop your model, the T icon displays the transformation dependencies that exist between an element in a PIM and elements in the PSMs</p>	<p>Example diagram ^[505]</p> <p>The Traceability Window ^[495]</p> <p>Model Search ^[475]</p> <p>The Project Browser ^[441]</p>
Relationship Matrix	<p>Using the Relationship Matrix, you can both create and study the relationships between, for example, the Requirements and Use Cases for a module</p> <p>You might identify the 'theme' package (for example, <i>Manage Users</i>) in the Requirements model and in the Use Case model as the source and target packages, and explore the likely element and connector types in the packages; this, like the Traceability diagram, identifies which Requirements are (or should be) realized by which Use Cases</p> <p>You can then perform similar checks with the <i>Manage Users</i> packages in, say, the Use Case and Implementation models</p> <p>The Source and Target field browsers (...) enable you to examine child</p>	<p>Relationship Matrix ^[496]</p>

Topic	Detail	See also
	packages within the 'theme' package, and obtain further detail on how the feature at this stage is defined	
Relationships Window	The Relationships Window shows a single level of relationships from the currently selected element Each line corresponds to a single relationship and provides information about that relationship and the element at the other end of the relationship	The Relationships Window ^[504]
Dependency Report	The Dependency report provides a list of dependencies for all elements in the model	Dependency Report ^[1783]
Implementation Report	The Implementation report provides a list of all elements in the model that have to be implemented, and the elements that implement them	Implementation Report ^[1785]

4.3.2 The Traceability Window

The Traceability window enables you to quickly see how elements are connected and how they influence each other.

- When an element is selected, it immediately becomes the top point in the Traceability window
- When a diagram is selected, all elements in the diagram are listed in the Traceability window, enabling you to use any of them as a starting point.

Access: **View | Traceability (Ctrl+Shift+4)**

Use to:

- Locate related elements in the Project Browser
- View the properties of related elements
- Show the diagrams where related elements are used
- Add related elements to the current diagram
- View the source code for related elements
- Change the display using the window toolbar options below:

Option	Action	See also
Generalizations	Show Generalization connectors	
Aggregations	Show Aggregation and Composition connectors	
Nesting	Show Element nesting and Nesting connectors	
Realizations	Show Realization connectors	
Dependencies	Show Dependency connectors	
Transitions	Show State Transition, Control Flow, and Object Flow connectors	
Other Links	Show the following connectors: <ul style="list-style-type: none"> • Association • Use Case • Delegate 	

Option	Action	See also
	<ul style="list-style-type: none"> • Assembly • Deployment • Information Flow • Manifest 	
Classifiers	Show where an element is used as the classifier of another element	
Embedded Element Reuse	Show where a Port or Part is also represented in another part of the model	
Transformations	Show where an element is created by running a Transformation from another element	Model Transformation ^[130]
Custom References	Show the Custom References that have been added between elements	Set Up Cross References ^[630]
Qualified Names	Show element names including the names of owning objects	

Notes:

- Restricting the relationship types shown to the smallest set you are interested in makes it easier and faster to find the elements you are interested in
- As an alternative view or to include the information in documentation, the Traceability window aids you in creating a Traceability diagram

Learn More:

- [Create Traceability Diagrams](#) ^[505]

4.3.3 Relationship Matrix

The Relationship Matrix is a spreadsheet display of relationships between model elements within packages.

Topic	Detail	See also
Overview	<p>The Relationship Matrix is a convenient method of visualizing relationships quickly and definitively</p> <p>It also enables you to create, modify and delete relationships between elements with a single mouse click - another quick way to set up complex sets of element relationships with a minimum of effort</p> <p>On the Relationship Matrix, you select:</p> <ul style="list-style-type: none"> • A source package • A target package • The relationship type and • The relationship direction <p>Enterprise Architect identifies all the relationships between source and target elements by:</p> <ul style="list-style-type: none"> • Listing the source package elements down the side of the matrix • Listing the target package elements across the top of the matrix and • If a relationship exists between a source and target element, 	Include Other Users' Packages ^[289]

Topic	Detail	See also
	<p>highlighting the intersecting grid square and displaying an arrow indicating the direction of the relationship</p> <p>The direction is a reflection of which elements are the source elements and which the target (it does <i>not</i> indicate the Direction property of the connector, as defined in the connector Properties dialog); the display might also show a 'bent arrow' icon, indicating that connectors exist in both directions between the source and target elements</p> <p>The highlighted squares are normally green, indicating that the <i>source</i> element is not locked (because the parent package has not been checked in under version control); if the element <i>is</i> locked (the parent package has been checked in) the highlight is pink</p> <p>If you click on any square in the matrix, the square, the row header and the column header are highlighted</p>	
<p>Operations</p>	<p>Once you open the Relationship matrix, you can:</p> <ul style="list-style-type: none"> • Update, delete and create relationships through the Relationship Matrix • Select options for modifying the type of information the Relationship Matrix displays • Export the contents of the Relationship Matrix to a CSV file, a .png file or a .emf file • Print the contents of the Relationship Matrix, scaled down if required • Save a profile of the Relationship Matrix settings to monitor development of the same source and target packages • Investigate the Source and Target elements in the relationship 	<p>Open the Relationship Matrix ^[497]</p> <p>Creating and Deleting Relationships ^[500]</p> <p>Relationship Matrix Options ^[501]</p> <p>Matrix Profiles ^[499]</p> <p>Review Source and Target Elements ^[503]</p>

4.3.3.1 Open the Relationship Matrix

Access: **View | Relationship Matrix**
Project Browser Package Context Menu | Documentation | Open in Relationship Matrix | As Source or | As Target

Once the Relationship Matrix opens you:

- [Select which element type to show](#) ^[497]
- [Select the connector type and direction to show](#) ^[498]
- [Set the source and target packages](#) ^[498]

The Relationship Matrix refreshes after every change you make to the input parameters.

4.3.3.2 Set Element Type

The Relationship Matrix defaults to show all element types, or you can specify which types to show.

How to:

To set the source and/or target element types, follow the steps below:

Step	Action	See also
1	Click on the drop-down arrow in the Type field for the source package, and/or for the target package	
2	Find the required element type in the list and click on it Enterprise Architect refreshes the Relationship Matrix content	

4.3.3.3 Set Connector Type and Direction

In the Relationship Matrix, you must set the connector type to report on and the connector direction.

How to:

To set the connector type and direction, follow the steps below:

Step	Action	See also
1	Click on the drop-down arrow in the Link Type field	
2	Find the required connector type in the list and click on it	
3	Click on the drop-down arrow in the Direction field	
4	Find the required direction in the list and click on it Enterprise Architect refreshes the Relationship Matrix content <ul style="list-style-type: none"> If you set Direction to Both, each relationship is indicated by two arrows - a <i>From-To</i> arrow and a <i>To-From</i> arrow The direction is a reflection of which elements are the source elements and which are the target.; it does not indicate the Direction property of the connector, as defined in the connector Properties dialog 	

4.3.3.4 Set Source and Target Package

You must set both the source and target packages for the Relationship Matrix before relationships can be displayed.

You set the source and target packages AFTER setting the connector and element types/details; as Enterprise Architect refreshes the content after each change, this is usually faster.

The Relationship Matrix includes ALL child elements in a hierarchy. Sometimes in a large model this can be a lot of elements, possibly too many to be useful. Take care in selecting the source and target package.

How to:

To set the source and target packages, follow the steps below:

Step	Action	See also
1	In the Project Browser, click on the required source package, then press (Ctrl) and click	

Step	Action	See also
	on the required target package, to select the two packages together	
2	<p>Drag the selected packages over the Source and Target fields</p> <p>The first-selected package name displays in the Source field, and the second-selected package name displays in the Target field</p>	
<i>Alternatively</i>		
1	<p>Select and drag a single package name over the Source OR Target field, to change just the source or the target package</p> <p>If you drop the package name anywhere else on the Relationship Matrix, the system prompts you to specify whether to add it to the Source or Target field</p>	
<i>Alternatively</i>		
1	<p>Click on the (...) (Browse) button at the end of the Source or Target field</p> <p>A short menu displays, with options to:</p> <ul style="list-style-type: none"> • Locate a package using the Browse Project dialog or • Run a Model Search (selected from a drop-down list) on a search term, on whatever package is selected in the Project Browser; the name of the search is displayed in the Source or Target field <p>If required, you can browse for a package in one field and run a search in the other</p>	
2	Select the required package, or the required Model Search and search term, and click on the OK button	

In all cases, Enterprise Architect immediately:

- Populates the Relationship Matrix axes with the elements identified in the Source and Target packages or the searches that meet the selection criteria, and
- Shows any relationships between the sets of elements that also meet the selection criteria.

4.3.3.5 Matrix Profiles

If you use a particular combination of field values often, you can save that combination as a profile to make it easier to recall them for use.

How to:

To save a certain Relationship Matrix configuration as a named profile for later recall, follow the steps below:

Step	Action	See also
1	<p>Set up the Relationship Matrix with the required:</p> <ul style="list-style-type: none"> • source and target packages or Model Searches • source and target element types and • connector type and direction 	<p>Set Element Type^[497]</p> <p>Set Connector Type and Direction^[498]</p> <p>Set Source and Target Package^[498]</p>
2	Click on the Options button on the Relationship Matrix and select the Profiles Save as New Profile menu option	

Step	Action	See also
	A short dialog displays	
3	In the Enter Value field, type a profile name of up to 12 characters Click on the OK button	
4	Once you have created a profile, you can select it by clicking on the Profile drop-down arrow at the top of the Relationship Matrix screen and selecting the profile name from the list	

Notes:

- You can modify an existing profile by:
 - Selecting the profile name from the **Profile** drop-down list
 - Selecting the **Options: Profiles | Update Current Profile** menu option
 - Changing the field values, then
 - Saving the profile under the same profile name
- To delete an existing profile, select it in the **Profile** drop-down list and select the **Options: Profiles | Delete Current** menu option

4.3.3.6 Creating and Deleting Relationships

From the Relationship Matrix, you can:

- Modify relationships
- Delete relationships
- Create new relationships

Use to:

- Quickly create and manage relationships like Realization and Aggregation between Requirements and implementation elements (such as Use Cases)

How to:

To modify or delete a relationship on the Relationship Matrix, follow the steps below:

Step	Action	See also
1	Right-click the required highlighted relationship square on the matrix, and select the appropriate context menu option: <ul style="list-style-type: none"> View relationship - opens the Properties dialog for the selected relationship Source element properties - opens the Properties dialog for the source element Target element properties - opens the Properties dialog for the target element Delete relationship 	
2	If you have selected Delete relationship , Enterprise Architect prompts you to confirm this action Click on the Yes button The Delete relationship option is not available if:	

Step	Action	See also
	<ul style="list-style-type: none"> The source element (that is, the owner) is locked You have selected Both in the Direction field - you are effectively trying to delete half a relationship 	
3	<p>If you have selected one of the other options, modify the properties in the dialog fields as required</p> <p>Click on the OK button to save the changes</p>	

To create a new relationship, follow the steps below:

Step	Action	See also
1	In the Link Type field, select the required relationship type	
2	Right-click on the empty square at the intersection of the appropriate source row and target column	
3	<p>Select the Create new relationship context menu option</p> <p>A submenu displays, listing the types of relationship you can create</p>	
4	Click on the required type of relationship; a connector of that type is created between the two elements	

4.3.3.7 Relationship Matrix Options

The Relationship Matrix provides a number of menu options for modifying the output of information, for preserving the settings for such modifications, and for capturing the output.

Access: [View | Relationship Matrix: Options](#)

Use to:

- Output the information on the Relationship Matrix to the printer or to a metafile, .png file or .csv file
- Create and update profiles of the configurations of the matrix that you have designed
- Define local settings to control what the Relationship Matrix displays

How to:

Topic	Detail	See also
Print Relationship Matrix	<p>Click on the Matrix Print menu option</p> <p>The Print dialog displays, on which you select the output printer and specify the printer properties, the range of pages to print, and the number of copies</p> <p>The output is a WYSIWYG representation of the Relationship Matrix contents</p>	

Topic	Detail	See also
Display a preview of printout	<p>Click on the Matrix Print Preview menu option</p> <p>The Print Preview screen displays, showing the Relationship Matrix printout</p>	Print Preview [540]
Scale printout	<p>When you print the Relationship Matrix, by default it prints on as many pages wide and long as the matrix requires</p> <p>You can scale the printout into a fixed number of pages wide, and the row height is automatically adjusted to maintain the proportions of the matrix; this reduces the overall size of the printout and improves appearance, especially when used in conjunction with the Landscape option in the printer properties</p> <p>For example, a 16-page printout without scaling can, with a scaling of 2 pages wide, be reduced to 6 pages</p> <p>To set the page scaling:</p> <ol style="list-style-type: none"> 1. Click on the Matrix Scale Setting menu option The Scale Matrix dialog displays 2. Select the Scale Matrix Width Into Pages checkbox, and type or select the number of pages in width to scale to 3. Click on the OK button to apply the setting 	
Save Relationship Matrix as graphic file	<p>Click on the Matrix Save as Metafile or Matrix Save as PNG menu options</p> <p>A Browser dialog displays that enables you to select the target file location and specify the filename of the .emf or .png file in which to save the output</p> <p>You can incorporate these files in an RTF or HTML report, as a hyperlinked file or an included file</p>	
Export output to CSV file	<p>The contents of the Relationship Matrix can be exported to a CSV file, which provides a convenient mechanism for moving the matrix data to a spreadsheet environment such as Microsoft Excel</p> <p>This option is also active in the 'Lite', read-only version of Enterprise Architect</p> <p>To export to CSV:</p> <ol style="list-style-type: none"> 1. Select the Matrix Export to CSV menu option The Windows Browser dialog displays 2. Browse to the required file location and type in a .CSV filename to export to 3. Click on the Save button to export the data 	The Read-only 'Lite' Edition [18]
Create and manage matrix profiles	<p>If you use a particular combination of field values often, you can save that combination as a profile to make it easier to recall them for use</p>	Matrix Profiles [499]
Manage display content	<p>Select the Options menu option</p> <p>The Matrix Options dialog displays; select one or more of the following checkboxes as required:</p> <ul style="list-style-type: none"> • Include Source Children - to recursively include child packages and contents under the Source • Include Target Children - to recursively include child packages and contents under the Target • Include All Extended Meta Types - to include elements that are 	

Topic	Detail	See also
	<p>extensions of a specified meta-type; for example, if there are Block elements (extending Class) in the package, selecting this option and specifying the type Class includes Class and Block elements, and any further derivatives of Block in the matrix</p> <ul style="list-style-type: none"> • Sort Axes - to ensure package elements display in alphabetical order • Show Package Names - to hide or show package names in the Relationship Matrix; this is useful for shortening the displayed texts, especially in circumstances where packages have long names • Use Element Alias If Available - to display an element's alias instead of the element name, if an alias has been defined • Show Level Numbering If Available - to reproduce level numbering in the Relationship Matrix, if it is turned on in the Project Browser 	

4.3.3.8 Review Source and Target Elements

As well as providing information on connectors and relationships, the Relationship Matrix enables you to obtain information on the source and target elements in a relationship.

Topic	Detail	See also
Identify elements in relationships with a source or target element	<p>Click on the source or target element name in the row or column titles</p> <p>The entire row or column is highlighted</p> <p>Scroll across or down the highlighted row or column and quickly identify where the relationships are; this is very useful if the row or column is long</p>	
Display the Properties dialog for the selected element	<p>Right-click on the element name and select the Properties option</p>	
Identify diagrams in which the element is used	<p>Right-click on the element name and select the Find in Diagrams option</p> <p>Either:</p> <ul style="list-style-type: none"> • The only diagram in which the element is used displays, with the element highlighted, or • A list of the diagrams in which the element is used displays; you then double-click on the required diagram to open it 	
Highlight the element name in the Project Browser	<p>Right-click on the element name and select the Locate in Project Browser option</p> <p>The Project Browser expands to the location of the element, and the element name is highlighted</p>	
Make the element the focus in any docked screens or windows that are open	<p>Right-click on the element name and select the Set Context Item option; the selected element becomes the subject of any open windows or screens</p>	

4.3.4 The Relationships Window

The Relationships window displays all connectors between an element selected from a diagram or the Project Browser, and other elements. This provides a quick overview of an element's relationships in the model.

Access: [View | Relationships](#)

Topics:

Topic	Detail	See also
Content	<p>The Relationships window shows, for each connector, the:</p> <ul style="list-style-type: none"> • Connector type • Any stereotype on the connector • The Source and Target elements in the relationship • The direction of the relationship • Any stereotype on the Target element • The Target element type • The Target and Source element roles, if any <p>It also has a Target in Diagram column which, if the Target element for a connector is visible in the currently loaded diagram, contains the value Yes; this is very useful when you are dragging related elements from the Relationships list onto the current diagram</p> <p>Adding the target element to the diagram does not change the context to that element; the originally-selected element (the subject of the Relationships window) remains in context, and you continue working with that element</p>	
Facilities	<p>Double-click on a connector in the list to display the <connector type> Properties dialog, on which you can edit the connector attributes</p> <p>Right-click on a connector to open the context menu, which provides options to:</p> <ul style="list-style-type: none"> • locate the related element in the Project Browser • view the related element properties • delete the connector • hide certain connectors in diagrams • show hidden connectors • locate the selected element in any other diagrams in which it occurs • place the related element in the current diagram (if it is not currently visible in the diagram); this is useful in building a picture of what an element interacts with, especially when reverse engineering an existing code base 	

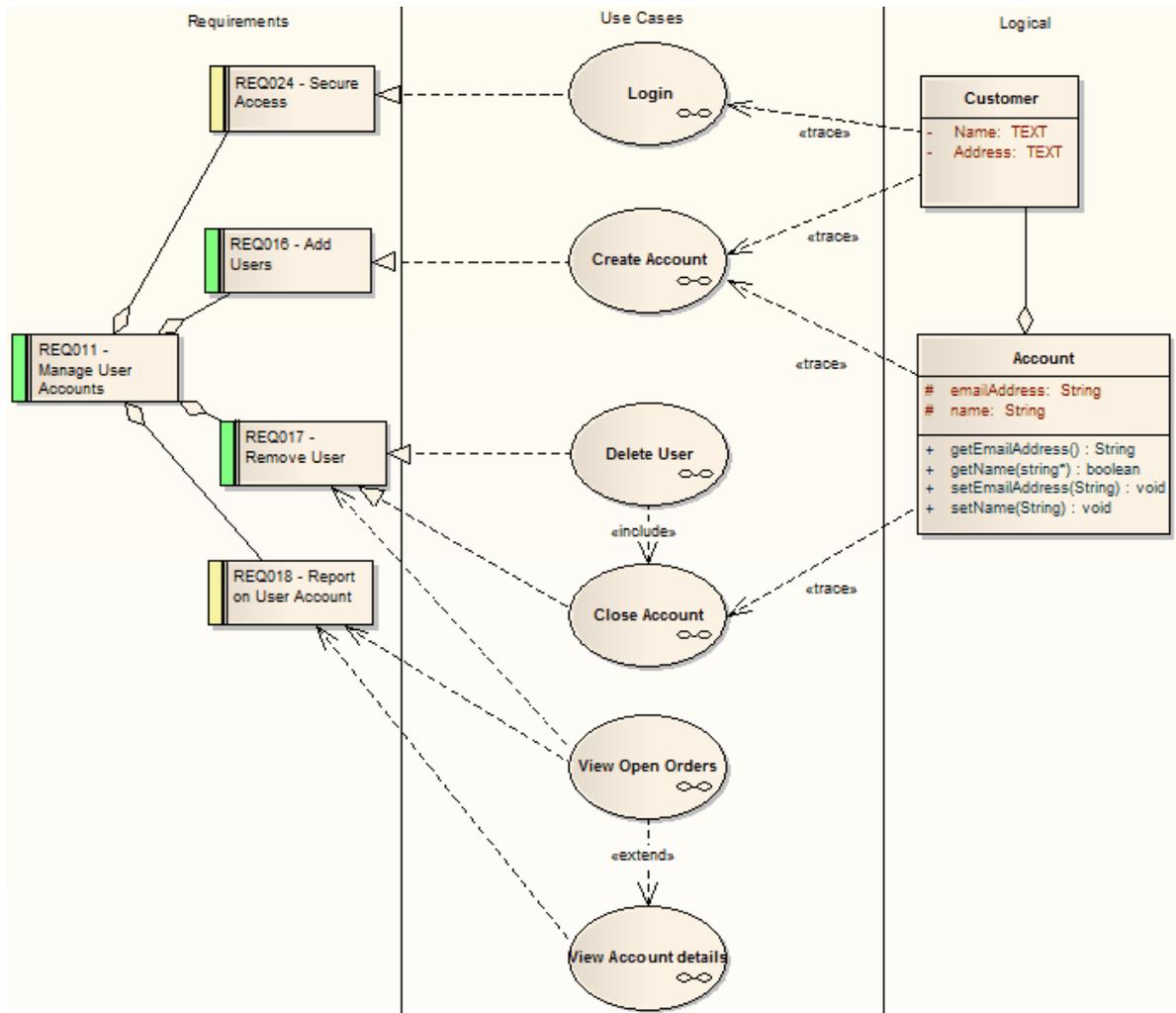
Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with security on, the diagram and the source and target elements must be free for editing before some of these options are available for use

4.3.5 Example Diagram

A wide variety of relationships can be shown on diagrams; you are not restricted to having a single element type or elements from a single phase on a diagram. This can help you to create appropriate relationships (such as Realize or Trace) to elements where no relationship was previously defined.

Consider the following diagram:



Topic	Detail	See also
Diagram Contents	<p>The example diagram above shows how a series of requirements are traced to Use Cases and then Classes</p> <p>The diagram instantly shows:</p> <ul style="list-style-type: none"> • That two levels of Requirements are realized by Use Cases • Which Requirement is realized by which Use Case(s) • How some of the Use Cases are implemented by Class elements <p>You can drill down on each Use Case (or, in other Traceability diagrams, any other composite element) to display more detailed diagrams showing how the Use Case meets the Requirement; the Close Account Use Case, for example, contains a Communication diagram and a Sequence diagram.</p>	

Topic	Detail	See also
	<p>You can tailor your Traceability diagrams to depict any level of granularity and any stages of development that are appropriate; for example, you might:</p> <ul style="list-style-type: none"> • Narrow the above diagram to show development from just the Remove User Requirement, or • Extend it to include Interfaces, Components, Test Case elements or any other facet of the system or process 	
Building a Diagram	<p>A number of tools are available to for creating diagrams that show traceability:</p> <ul style="list-style-type: none"> • The diagram context menu Add Related Elements option enables you to automatically bring in elements linked to the selected element • The Relationships window enables you to add elements to the diagram that are directly linked to the currently selected element • The Traceability window enables you to add elements that are indirectly linked 	Insert Related Elements <small>[652]</small>

Notes:

- Whilst the Traceability diagram itself provides information on the definition, design and implementation of a business process feature, much more information can be obtained using the Traceability tools

Learn More:

- [Realize](#) [1007]
- [Trace](#) [1012]
- [Traceability Tools](#) [494]
- [The Relationships Window](#) [504]
- [The Traceability Window](#) [495]

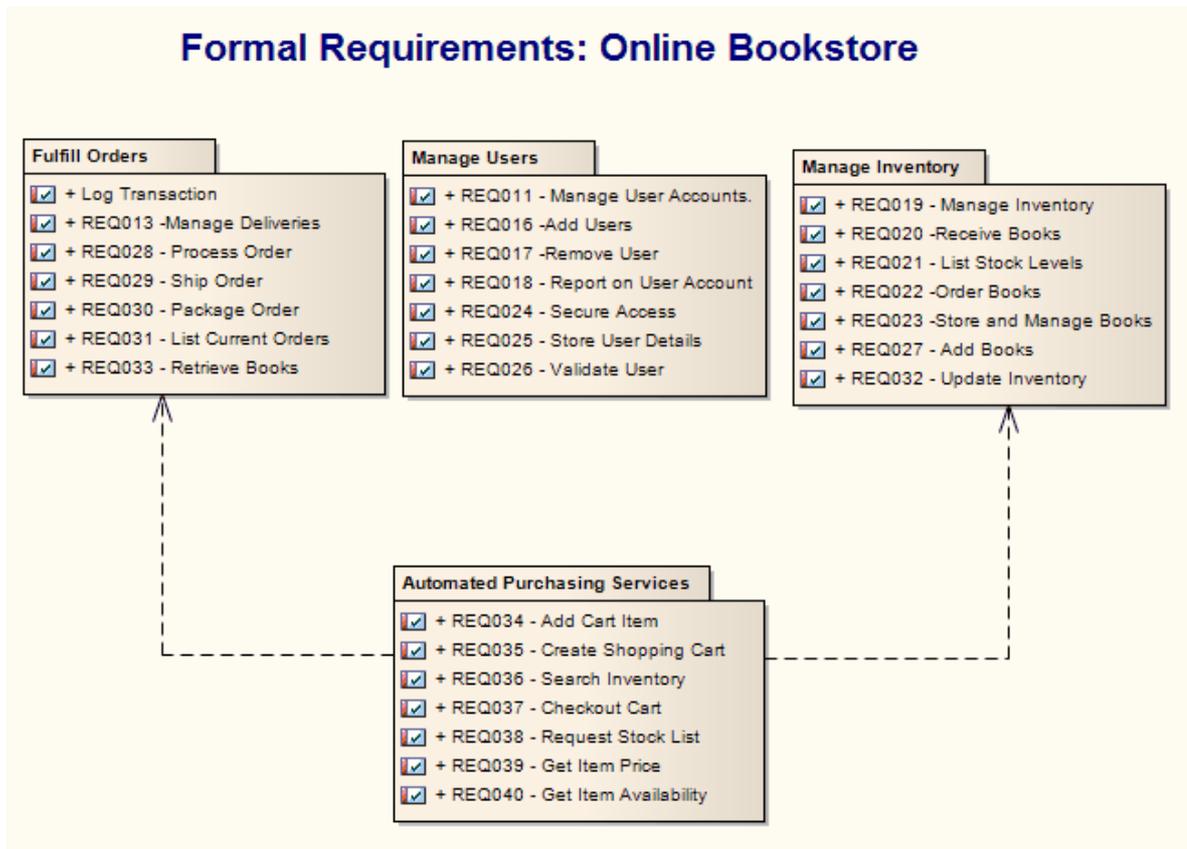
4.3.6 Packages and Elements

Analyzing the requirements of a system or process helps you identify units of functional activity in that system or process. You can represent these units with Packages, which you then populate with elements of relevance to the functional units and to the stage of development.

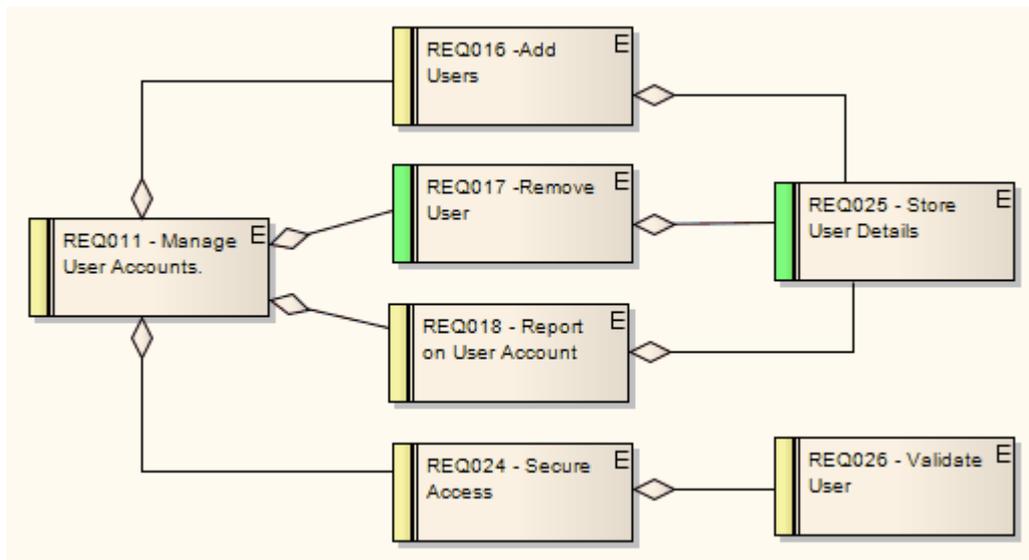
In the following four diagrams, you can see how Requirements and Use Case Packages are created to group the Requirements elements and Use Case elements that define and start to implement each of several functional areas.

By maintaining the same structure in each model in the project you also make it easy to trace the project development through the stages.

Requirement Packages for Online Bookstore Process:

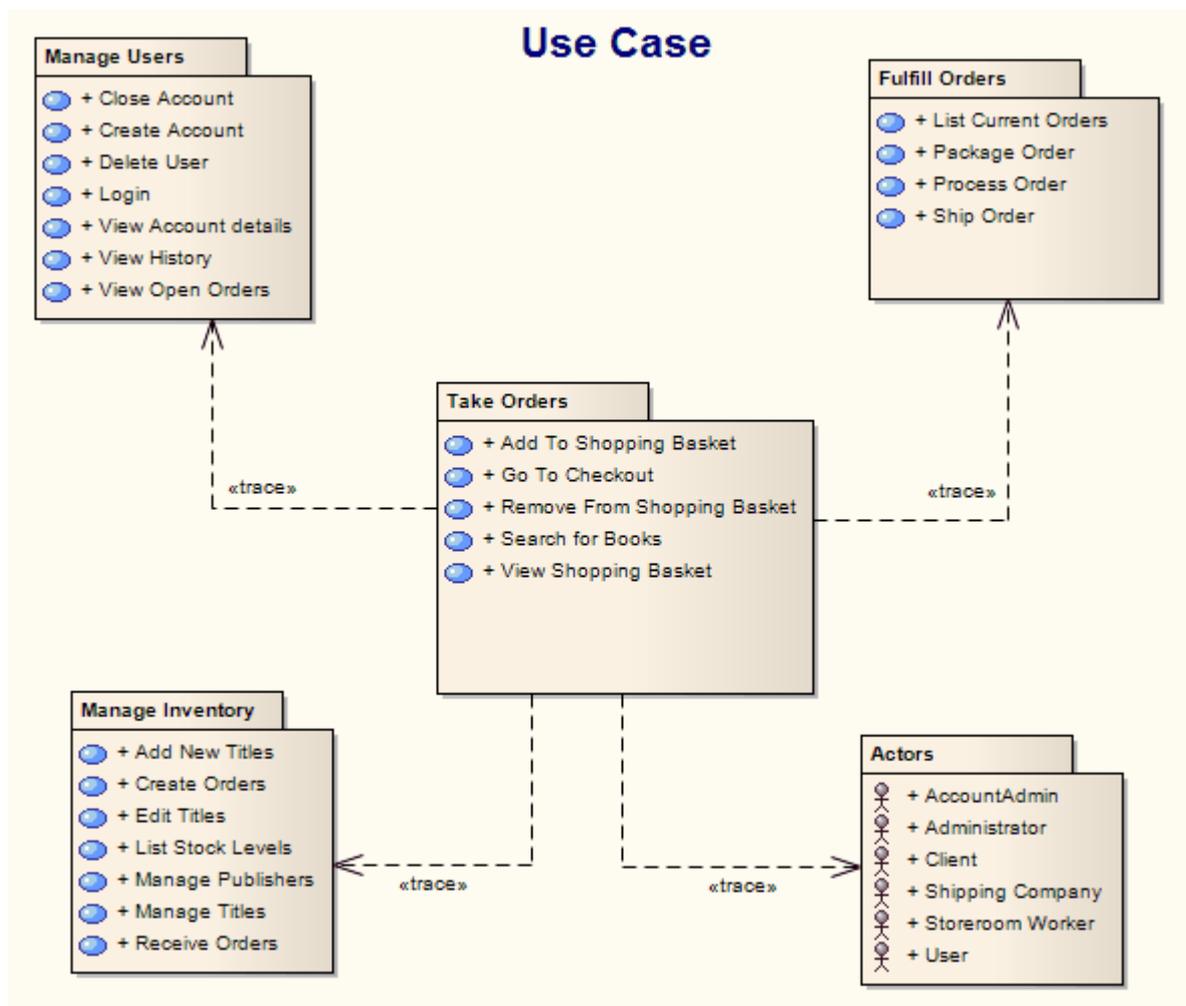


Requirements in Manage Users Unit of Online Bookstore Process:

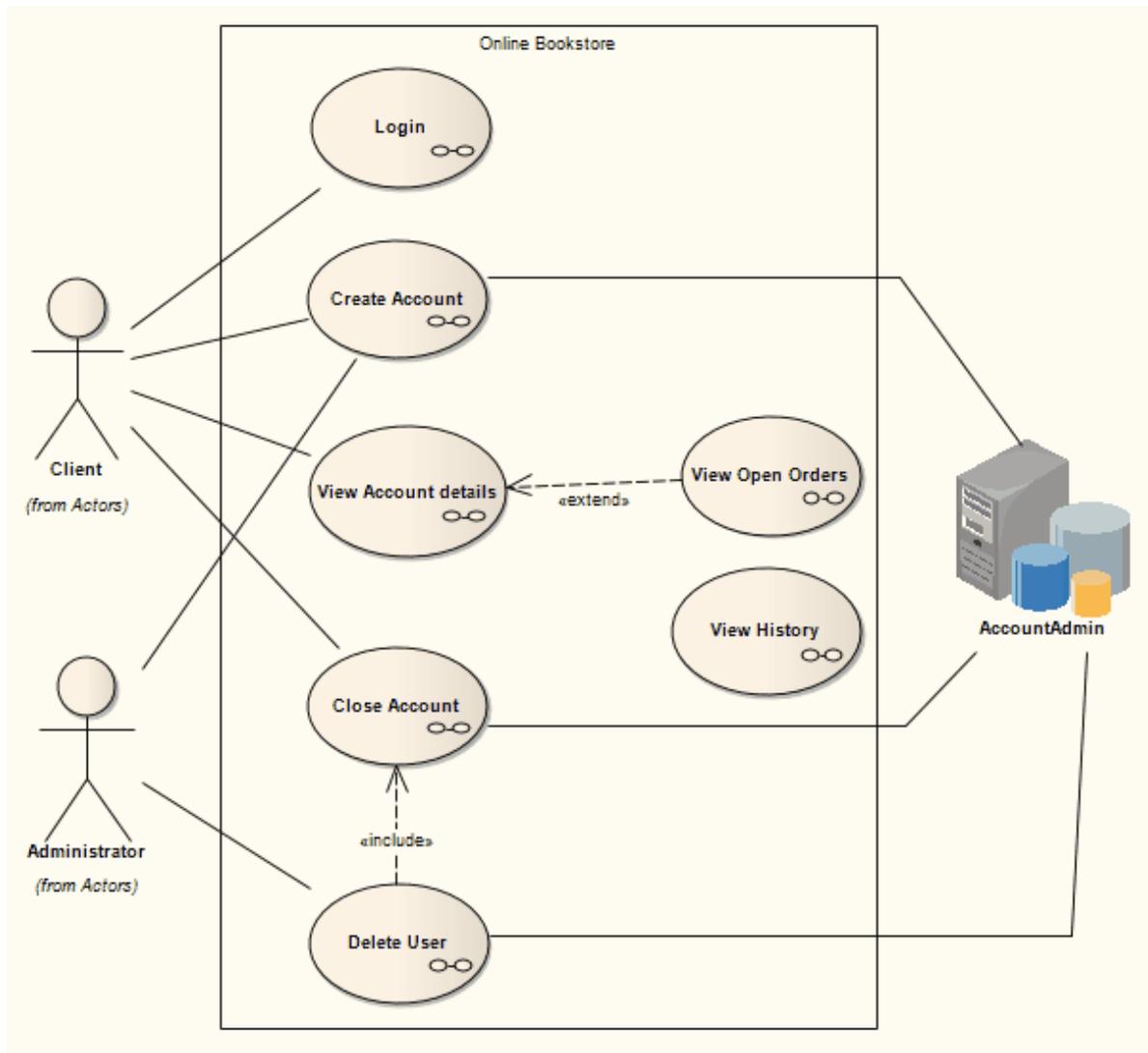


The Requirements diagram also makes it clear what Requirements form subsets of others, or are components of more than one other Requirement.

Use Case Packages For Online Bookstore Process:



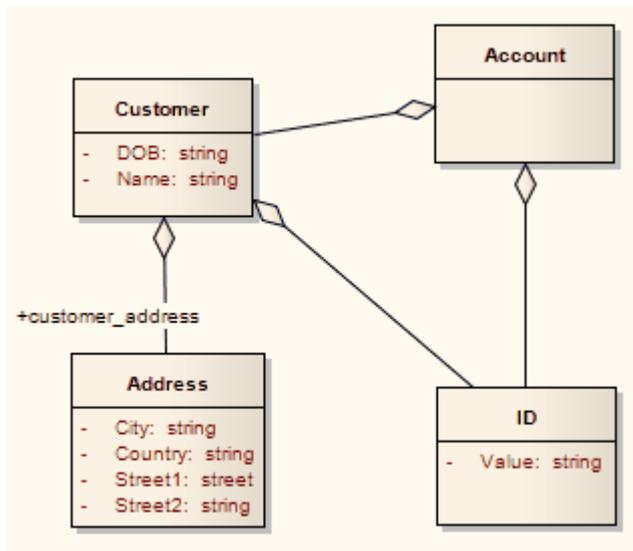
Use Cases in Manage Users Unit of Online Bookstore Process:



The Use Case diagrams can also clarify what aspects of a process require or enable human intervention, and which require or enable system intervention.

Implementation Stage:

For completeness, you could also consider the next stage, the implementation of some of these Use Cases, as represented by Class elements associated with this functional unit.

**Learn More:**

- [Trace: Tracking Dependencies](#) ⁴⁹³

4.3.7 Gap Analysis Matrix

The Gap Analysis Matrix is a tool to analyze model artifacts for potential gaps in solutions.

The idea is to highlight a shortfall between the Baseline Architecture and the Target Architecture; that is, items that have been deliberately omitted, accidentally left out, or not yet defined.

Access: [View | Gap Analysis Matrix](#)

Topic	Detail	See also
Overview	<p>The Gap Analysis Matrix is a convenient method of identifying gaps between solution architectures</p> <p>It enables you to model the gaps in the repository, which can later be addressed and assigned as tasks; the identified gaps can also be used to prioritize activities</p> <p>On the Gap Analysis Matrix, you select the:</p> <ul style="list-style-type: none"> • Target Architecture package • Baseline Architecture package • Type of Target Architecture artifact • Type of Baseline Architecture artifact • Type of element to model the gap 	Include Other Users' Packages ²⁸⁹
Operations	<p>Once you open the Gap Analysis Matrix, you can:</p> <ul style="list-style-type: none"> • Create, update and delete Gap Matrix Profiles to monitor development of the Baseline and Target packages • Edit Gap notes during a discussion session for Gap analysis • Create elements to address gaps 	Edit Gap Notes ⁵¹²

Topic	Detail	See also
	<ul style="list-style-type: none"> • Add links to existing elements to address gaps • Remove links to existing elements modeled as gap 	
Select Package	Click on the (...) (Browse) button at the end of the Target Architecture or Baseline Architecture field The Browse Project dialog displays Select the required package and click on the OK button; by default all the elements under the package are displayed	
Set Element Type	Click on the drop-down arrow in the Filter field for the Target Architecture package, and/or for the Baseline Architecture Find the required element type in the list and click on it; Enterprise Architect refreshes the matrix content	
Set Gap Type	Click on the drop-down arrow in the Record Gap As field Find the required element type in the list and click on it When you create a gap, the specified type of element is created; for example, if you want to address the gap with an issue element then select Issue for this field	
Baseline Architecture	The Baseline Architecture artifacts are listed along the rows of the matrix The last row is for capturing the new elements in the target architecture along the columns but not in the baseline architecture, hence the name New	
Target Architecture	The Target Architecture artifacts are listed along the columns of the matrix The last column is for capturing the missing or intentionally eliminated elements in the baseline architecture along the rows but not in the target architecture; hence the name Missing / Eliminated	
Modeling Gap	Right-click on a cell in the New row or Missing / Eliminated column The context menu displays options to: <ul style="list-style-type: none"> • Create a Gap element • Add a link to an existing Gap element If a link to Gap element already exists in the cell then the following options are displayed: <ul style="list-style-type: none"> • Edit Gap element • Find in Project Browser... • Remove Gap element link 	
Create Gap	Right-click on the cell and select the Create Gap Element context menu option The Browse Project dialog displays Select the package in which to create the Gap element and click on the OK button; a Gap element is created in the selected package and its Properties dialog displays to enable you to enter the element name and other required properties	
Link to Existing Gap	If you intend to use a Gap element that is already available in the model, right-click on the appropriate cell in the Missing / Eliminated column or New row and select the Link to Existing Gap Element context menu option	

Topic	Detail	See also
	The Select Classifier dialog displays, enabling you to select the existing Gap element	
Remove Link to Gap	<p>If you intend to remove a link to the Gap element in a cell, right-click on the appropriate cell in the Missing / Eliminated column or New row and select the Remove Link to Gap Element context menu option</p> <p>The link is removed from the cell but the element still exists in the Project Browser</p>	
Review Gap Element	<p>Right-click on the appropriate cell in the Missing / Eliminated column or New row and select the Edit Gap Element context menu option</p> <p>The Properties dialog displays, enabling you to edit the selected Gap element</p> <p>To locate the element in the Project Browser select the Find in Project Browser... context menu option, which highlights the element in the Project Browser</p>	
Gap Analysis Matrix Profiles	<p>On the Gap Analysis Matrix, you can create and manage profiles to save commonly-used combinations of Target Architectures, Baseline Architectures and stereotypes</p> <p>To work on Gap Analysis Matrix profiles, click on the Options button in the top right corner of the matrix; a submenu displays, listing options to:</p> <ul style="list-style-type: none"> • Create / Save the current matrix settings and content • Update the currently selected profile in the Profile field • Delete the currently selected profile in the Profile field 	
Open a saved Profile	<p>The Profile field drop-down list shows all the saved Gap Analysis Matrix profiles</p> <p>Click on the drop-down arrow in the Profile field, find the required Profile in the list and click on it to load the content of the selected profile in the matrix</p>	

Notes:

- The Gap Analysis Matrix is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

4.3.7.1 Edit Gap Notes

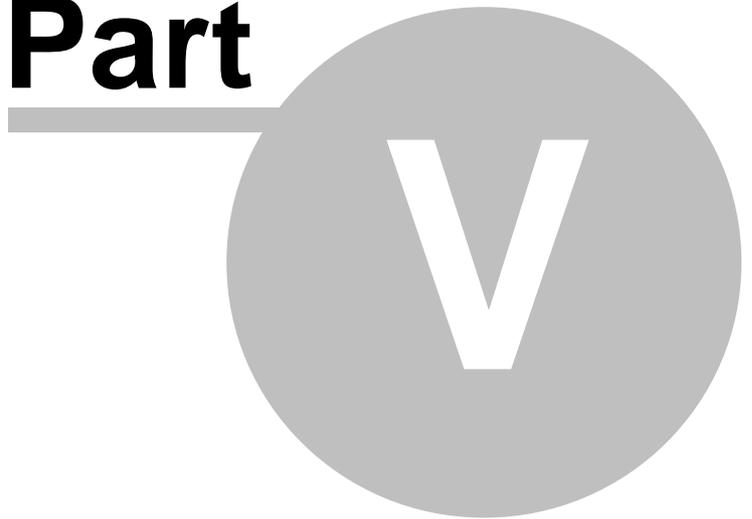
The Gap Analysis Matrix provides a way to record notes during the initial phase or during a discussion session for Gap Analysis. These notes are saved in profiles, which can later be reviewed and the appropriate model elements created to address these gaps.

How to:

To edit gap notes, follow the steps below:

Step	Action	See also
1	Right click on the cell at the intersection of a Target and Baseline element and select the Edit Gap Note option from the context menu Alternatively, double click on the cell The Gap Note dialog displays	Gap Analysis Matrix ⁵¹⁰
2	Edit the notes as required and click on the OK button	
3	The edited notes display in the selected cell in the matrix	
4	Update or save the matrix to retrieve it when the Gap Matrix profile is loaded at a later stage	

Part



5 Modeling Basics



Modeling can be described as graphically representing a business process or software system. The resulting model can be used to emphasize a certain aspect of the system being represented, and to record, document and communicate its detail. A study of such a model can enable insight or understanding of the system. Enterprise Architect's modeling platform is based on the Unified Modeling Language (UML), a standard that defines rules and notations for specifying business and software systems.

Using Enterprise Architect, you can quickly build a model using a hierarchy of *packages* to represent the structure and organization of the model.

Topics:

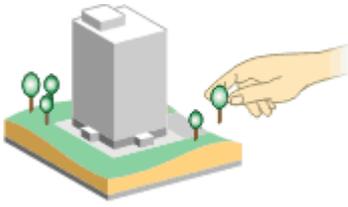
Topic	Link
<p>Models - a model is the highest conceptual level, representing a distinct and complete representation of all or some part of a modeled system</p> <p>A Project can contain multiple models.</p>	<p>Models ^[518]</p> <p>Model Wizard ^[518]</p> <p>Model Templates ^[519]</p>
<p>Views are the second level within a model and define a specific viewpoint of the system being modeled - for example a Use Case view, a Requirements View or a Dynamic (behavioral) View</p> <p>Views are simply packages which have an additional conceptual meaning.</p>	<p>Views ^[530]</p> <p>Add Views ^[531]</p>
<p>Packages are the basic containers that create the overall model structure. Packages hold other packages, elements, diagrams and similar model constructs.</p>	<p>Packages ^[532]</p> <p>Open Package in the Project Browser ^[532]</p>
<p>Diagrams are visual representations of how model elements are connected or behaviorally related. They can also display the characteristics of an element, such as attributes, methods, notes and tagged values in a convenient visual style.</p>	<p>Diagrams ^[538]</p> <p>Diagram Toolbox ^[546]</p> <p>The Quick Linker ^[622]</p> <p>Diagram Context Menu ^[538]</p> <p>Diagram Context Menu ^[538]</p> <p>Diagram Tabs ^[543]</p> <p>Diagram Tasks ^[567]</p> <p>Layout Diagrams ^[604]</p>
<p>Elements are the basic building blocks of models. They represent both structural constructs such as Classes and Interfaces, as well as behavioral constructs such as Activities, Actions and States.</p>	<p>Elements ^[625]</p> <p>Element Context Menu ^[648]</p> <p>Visual Representation ^[658]</p>
<p>Connectors are the various kinds of relationships between elements within a model - including behavioral relationships, associations,</p>	<p>Connectors ^[736]</p> <p>Connector Context Menu ^[736]</p>

taxonomic relations and similar.	Connector Tasks ^[740] Connector Properties ^[756]
Tagged Values are customizable, profile or user defined properties which are generally related to a Stereotype and define additional attributes and characteristics of an element. They are one of the fundamental means of extending UML into more domain specific areas.	Tagged Values ^[762] Quick Add - Tagged Value To Elements ^[763] Modify Tagged Values ^[766]
Notes are the standard common language based descriptions of what an element, diagram, feature or relationship is for and how it is used within the model. Notes are often used as a first step to eliciting the meaning and use of an element - which is later refined into something more concrete and precisely specified.	Notes ^[769] Notes Toolbar ^[770]
Reference Information - a variety of basic types and information that is used across a particular model - for example, Stereotypes, Tagged Values and similar.	Reference Data ^[772] UML Types ^[772]

Learn More:

- For information on UML, see the [Standard UML Models](#) ^[793] topic
- For examples of the UML models that Enterprise Architect can help you build, see the [Model Templates](#) ^[519] topic
- The [Quick Start Tutorial](#) ^[37] topic briefly shows you how to create a diagram within a package, containing elements and connectors
- Sparx Systems also provide a [Demonstration of quickly developing a Use Case model](#) (Online Resource)

5.1 Modeling



Enterprise Architect is a comprehensive UML analysis and design tool. It provides a library of UML data structures that you can use and extend to develop your models.

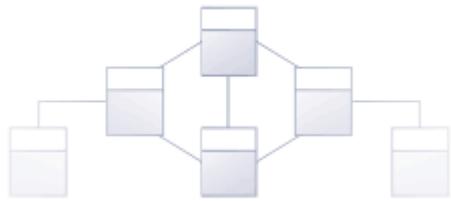
Topics:

Topic	Detail	See also
<p>UML Data Structures</p>	<ul style="list-style-type: none"> You create your projects and models using the Start Page or File Menu, which provide templates on which to base your models You initially create your packages and diagrams using the Toolbars and Menus, and the elements and connectors using the Diagram Toolbox You can also create new structures through the Project Browser, and re-use existing structures using the Project Browser, Model Views, Package Browser and Model Search <p>Building models requires the use of various UML data structures and Enterprise Architect tools, as above, to graphically represent a business process or software system. The resulting model can be used to emphasize a certain aspect of the system being represented and to record and communicate its detail.</p>	<p> The Start Page ^[70] File Menu ^[74] Model Templates ^[519] Workspace Toolbars ^[106] Main Menu ^[73] Diagram Toolbox ^[546] Project Browser ^[44] Model Views ^[464] Package Browser ^[456] Model Search ^[473] </p>
<p>Relationship Matrix</p>	<p>A further extremely useful tool is the Relationship Matrix, which enables you to visualize and amend the relationships and hence organization of structures within the model.</p> <p>Enterprise Architect also provides particular support for:</p> <ul style="list-style-type: none"> Requirements Management and Modeling the business process, an essential part of any software development process <p>You can extend the scope of your models by using UML Stereotypes, Profiles and Patterns, and MDG Technologies.</p>	<p> Modeling Fundamentals ^[515] Build Your Own Modeling Language ^[1038] Requirements ^[1152] </p>

Learn More:

- [Standard UML Models](#) ^[793]

5.2 Models



In Enterprise Architect a *model* is a special type of package, being the top level entry point to an Enterprise Architect project file.

Topics:

Topic	Detail	See also
Models	<p>You can develop a project with one model, or with several. Each model is a root node of a hierarchy of Model Packages and Views and, below them, packages. A model contains the diagrams, elements, relationships and associated metadata that define the structure and function of a system or process.</p> <p>These components are organized through the package hierarchy, which helps to group and manage related components. By iterating through all models, you can access all the elements within the project.</p> <p>You can create the model or models when you first create the project, or you can add and develop new models later. You can also delete a model, but be aware that everything contained in the model is deleted as well.</p>	

5.2.1 Model Wizard

The Model Wizard enables you to add models from a range of technologies to your new project, both on initial creation and over time.

Access: **Automatically displays when creating a new project; otherwise Right-click on a root node | Add | Add a New Model using Wizard (Ctrl) + (Shift) + (M)**

Use to:

- Add new models to a project

Reference:

Field	Usage	See Also
Technology	<p>If you have advanced Add-Ins and MDG Technologies, this panel lists them</p> <p>Click on the required technology to list the associated templates in the Name panel</p> <p>To list the standard Enterprise Architect model templates , select Basic UML 2 Technology</p>	

Field	Usage	See Also
Name	Click on the check box for each model type you want to create in your project	Model templates ⁵¹⁹
All	Select all of the models listed in the Name panel	
None	Clear all models selected	
OK	Create the Model Packages for your project	
Cancel	Abort the creation of model packages	
Help	Display this Help topic	

Notes:

- If you are a Technology Developer, you can also create and distribute custom templates as part of your own MDG Technology; the name of your technology displays in the Technology panel and, when you select the technology, the model template names display in the Name panel

Learn More:

- [Custom Templates](#) ¹⁰⁸⁷
- [MDG Technology](#) ¹⁰⁶⁴

5.2.2 Model Templates

The model templates contained in Enterprise Architect are designed to assist in the creation of projects and models for both new and experienced users.

Each template provides a framework on which you can create a model appropriate to your purpose, using the Model Wizard.

All model templates provided with Enterprise Architect follow the format described in the table below

Topics:

Topic	Detail	See also
Note	The note introduces you to the model template and outlines its purpose.	
Help Links	Help hyperlinks provide further information on how to use the model. Depending on the model template, links to examples and other useful information are also provided.	
Template	<p>The Template section in the model template provides a framework for creating your own model.</p> <p>Each template description provides an introduction to the terminology and icons used in the model templates, and gives a quick guide to the UML concepts important to the templates and how they are applied in Enterprise Architect.</p> <p>If you are a Technology Developer, you can also create and distribute custom templates as part of your own MDG Technology.</p>	<p>Business Process Model Template ⁵²⁰</p> <p>Requirements Model Template ⁵²¹</p> <p>Use Case Model Template ⁵²¹</p> <p>Domain Model Template ⁵²¹</p> <p>Class Model Template ⁵²³</p>

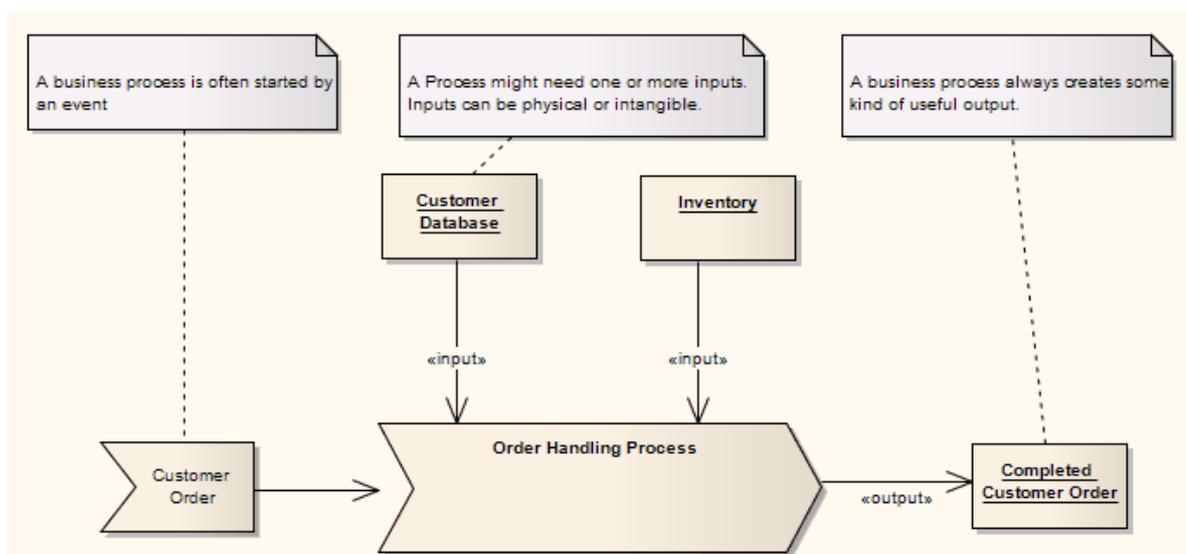
Topic	Detail	See also
		Database Model Template ^[524] Component Model Template ^[525] Deployment Model Template ^[526] Testing Model Template ^[527] Custom Templates ^[1087]

Learn More:

- [Model Wizard](#) ^[518]

5.2.2.1 Business Process Model Template

The *Business Process model* describes both the behavior and the information flows within an organization or system. As a model of business activity, it captures the significant events, inputs, resources, processing and outputs associated with relevant business processes.



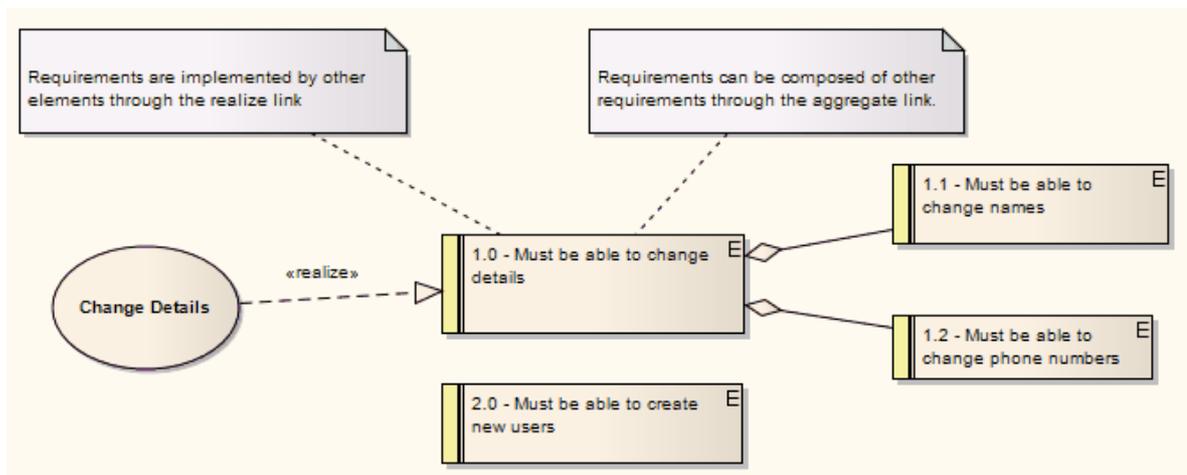
Learn More:

- [The Business Process Model](#)
- [Business Models](#) ^[1193]
- [Analysis Diagram](#) ^[1187]
- [Business Modeling/Interaction Diagrams](#) ^[1191]

5.2.2.2 Requirements Model Template

The *Requirements model* is a structured catalogue of end-user requirements and the relationships between them.

The Requirements Management facilities built into Enterprise Architect can be used to define Requirement elements, connect Requirements to other model elements, connect Requirements into a hierarchy and report on Requirements.



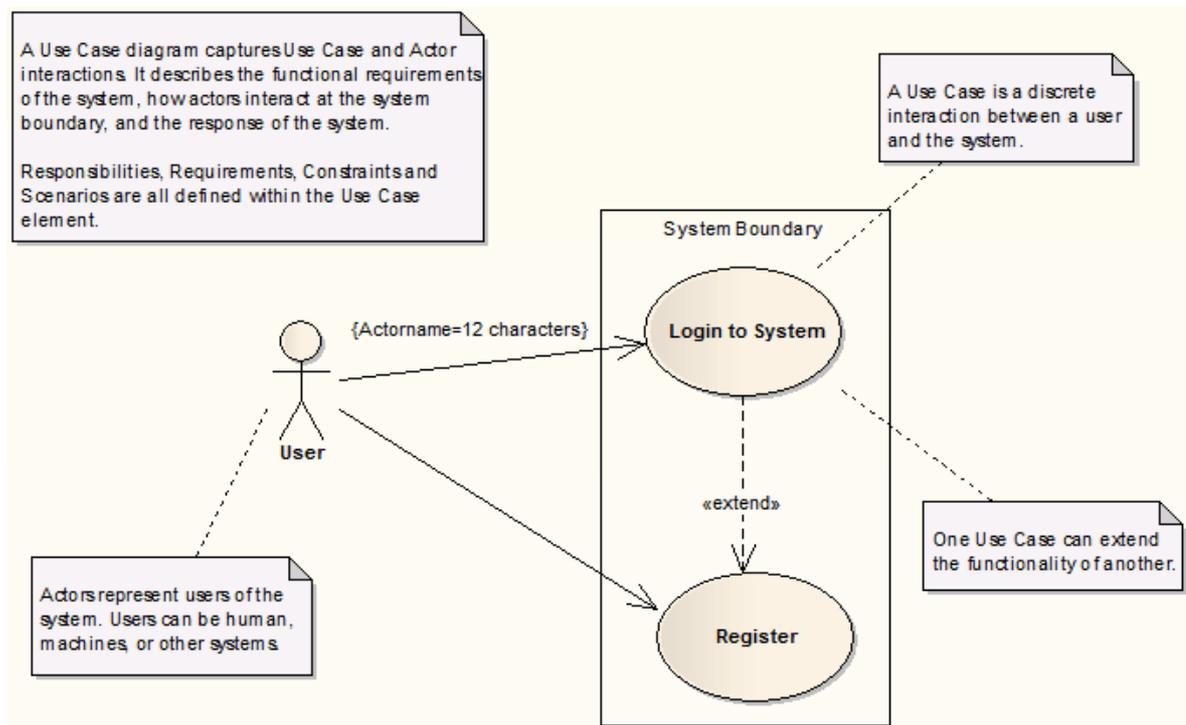
Learn More:

- [Requirements Management](#) ^[1152]
- [Requirements Management in Enterprise Architect](#)
- [Packages](#) ^[533]

5.2.2.3 Use Case Model Template

The *Use Case model* describes a system's functionality in terms of Use Cases.

Each Use Case represents a single repeatable interaction that a user or 'actor' experiences when using the system, emphasizing the user's perspective of the system and interactions.



Learn More:

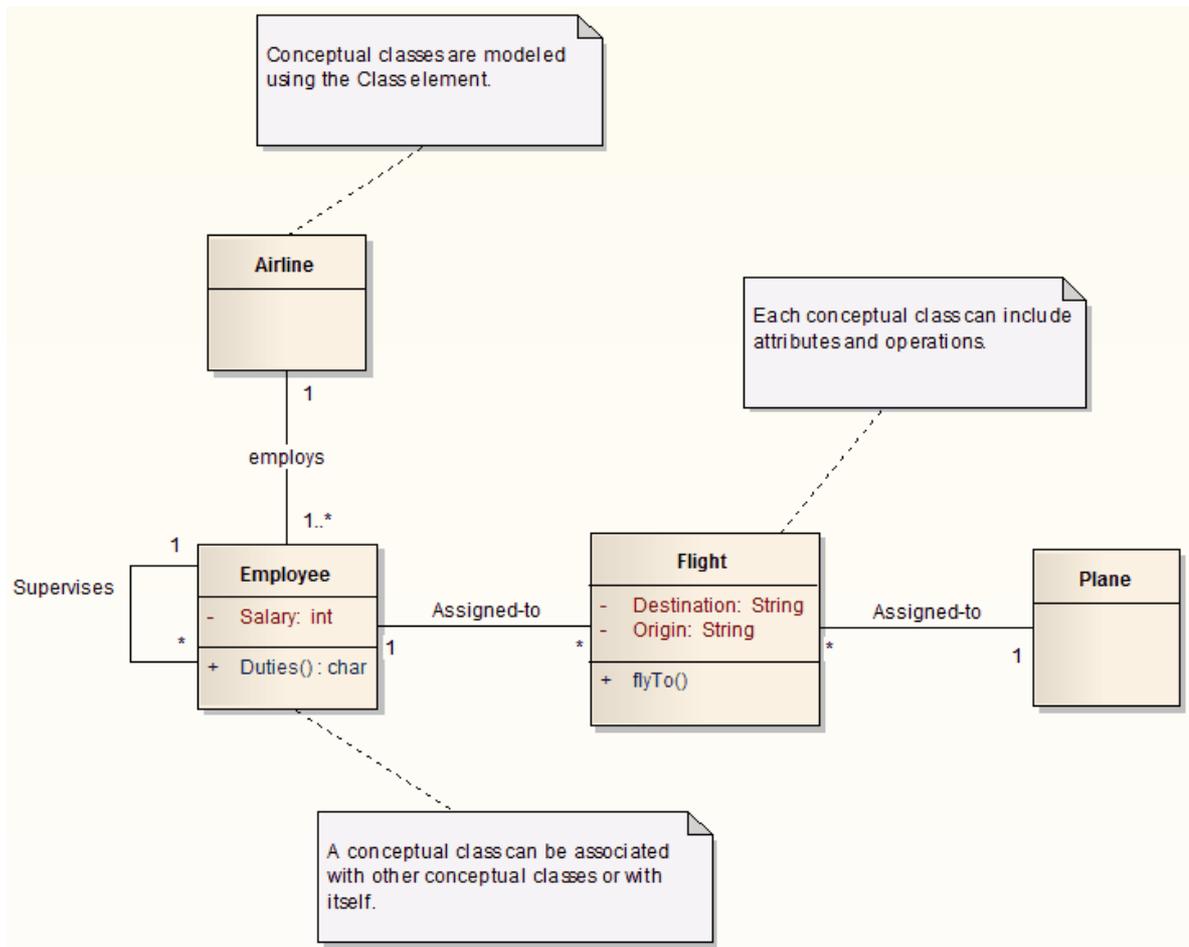
- [Use Case](#)^[935]
- [Use Case Diagram](#)^[813]
- [The Use Case Model](#)

5.2.2.4 Domain Model Template

A *Domain model* is a high-level conceptual model, defining physical and abstract objects in an area of interest to the Project.

The Domain model can be used to document relationships between and responsibilities of conceptual classes (that is, classes that represent the concept of a group of things rather than Classes that define a programming object).

It is also useful for defining the terms of a domain.



A Domain model shows:

- The physical and organizational units of the domain; for example, *Employee* and *Flight*
- The relationships between these units; for example, *Employee* is *assigned to* *Flight*
- The multiplicity of those relationships; for example, *one* employee can be assigned to *no* flights, *one* flight or *many* flights (represented by the 1 and the * at the ends of that relationship)

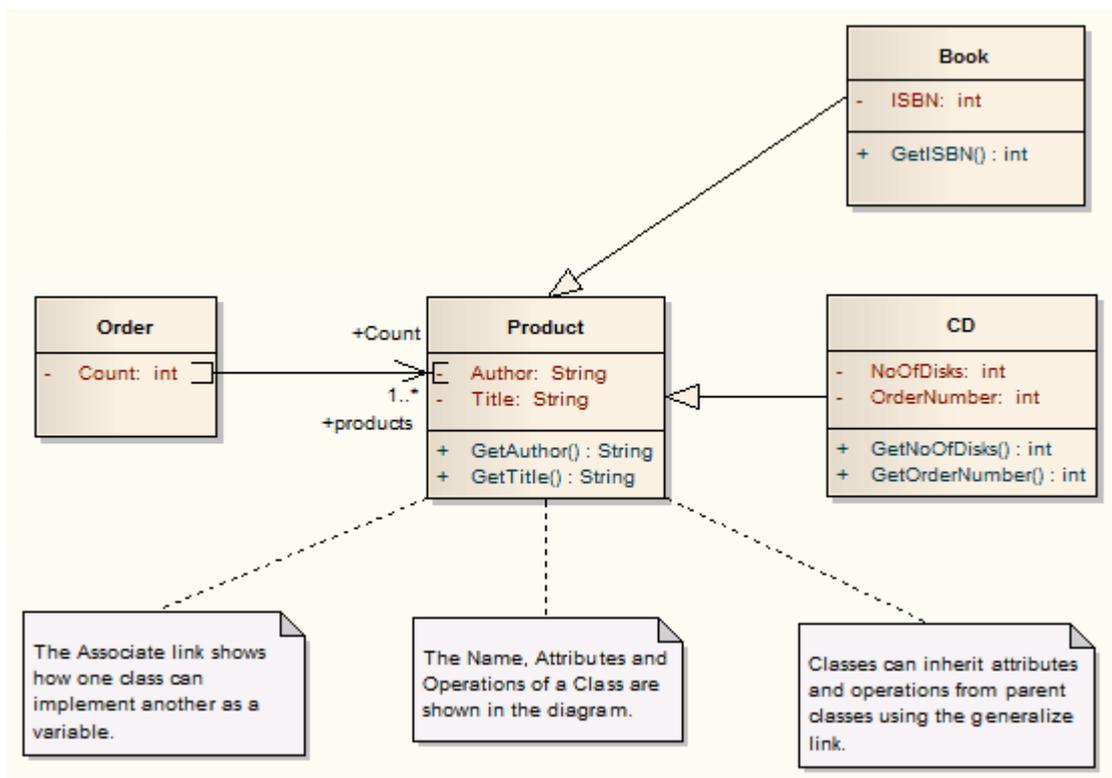
Learn More:

- [Multiplicity](#)^[759]

5.2.2.5 Class Model Template

The *Class model* is a rigorous, logical model of the software system under construction.

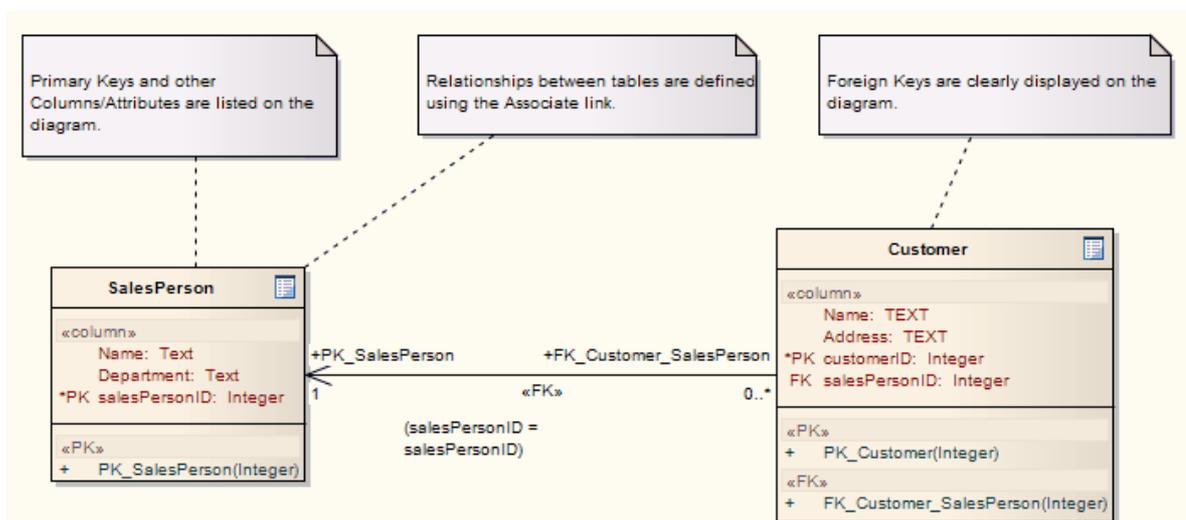
Classes generally have a direct relationship to source code or other software artifacts that can be grouped together into executable components.

**Learn More:**

- [Classes](#) ^[94]
- [Class Diagram](#) ^[798]
- [The Logical Model](#)

5.2.2.6 Database Model Template

The *Database model* describes the data that must be stored and retrieved as part of the overall system design. Typically this means relational database models that describe the tables and data in detail and enable generation of DDL scripts to create and set up databases.



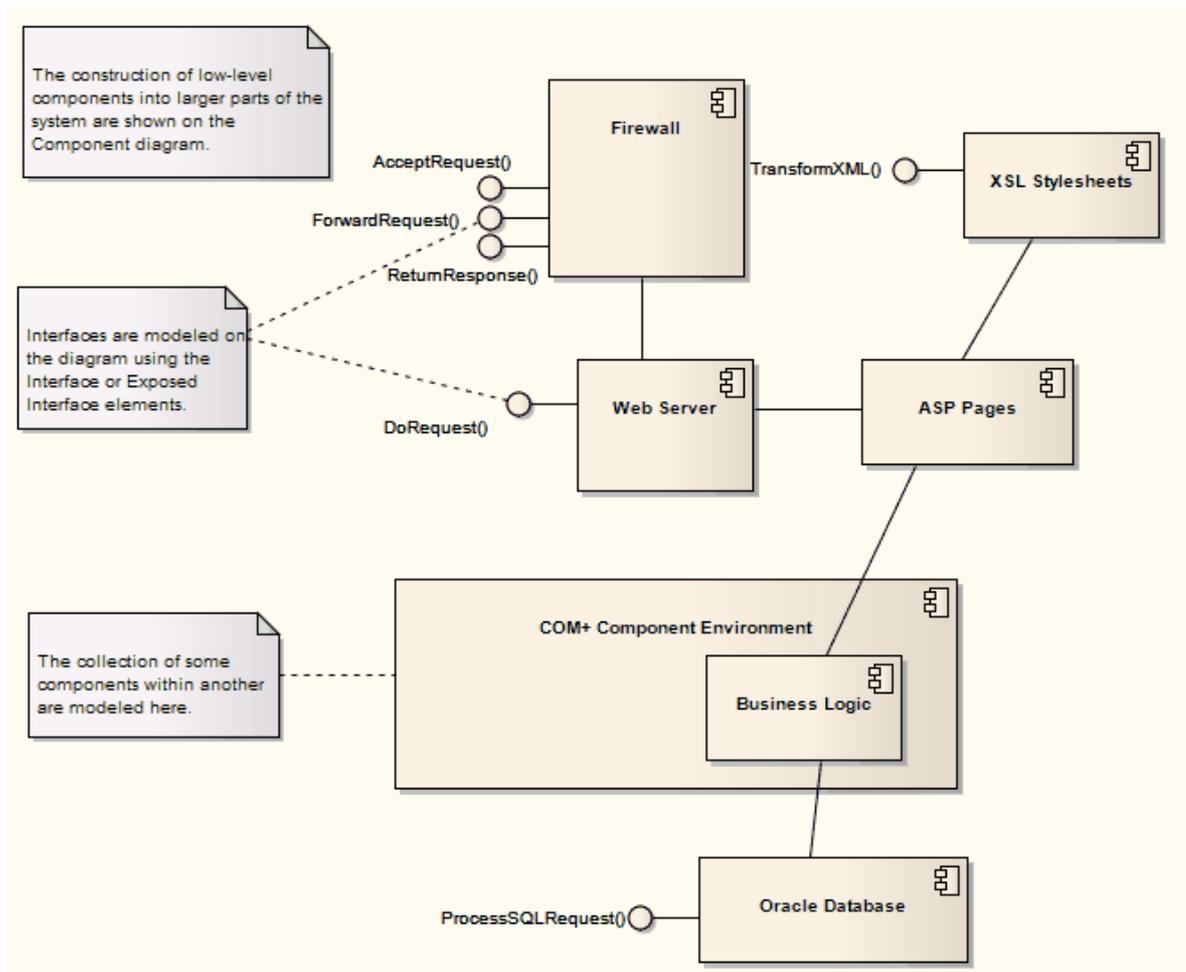
Learn More:

- [Data Models](#) ^[1276]
- [UML Database Modeling](#)

5.2.2.7 Component Model Template

The *Component model* defines how Classes, Artifacts and other low level elements are collected into high level components, and describes the interfaces and connections between them.

Components are compiled software artifacts that work together to provide the required behavior within the operating constraints defined in the Requirements model.

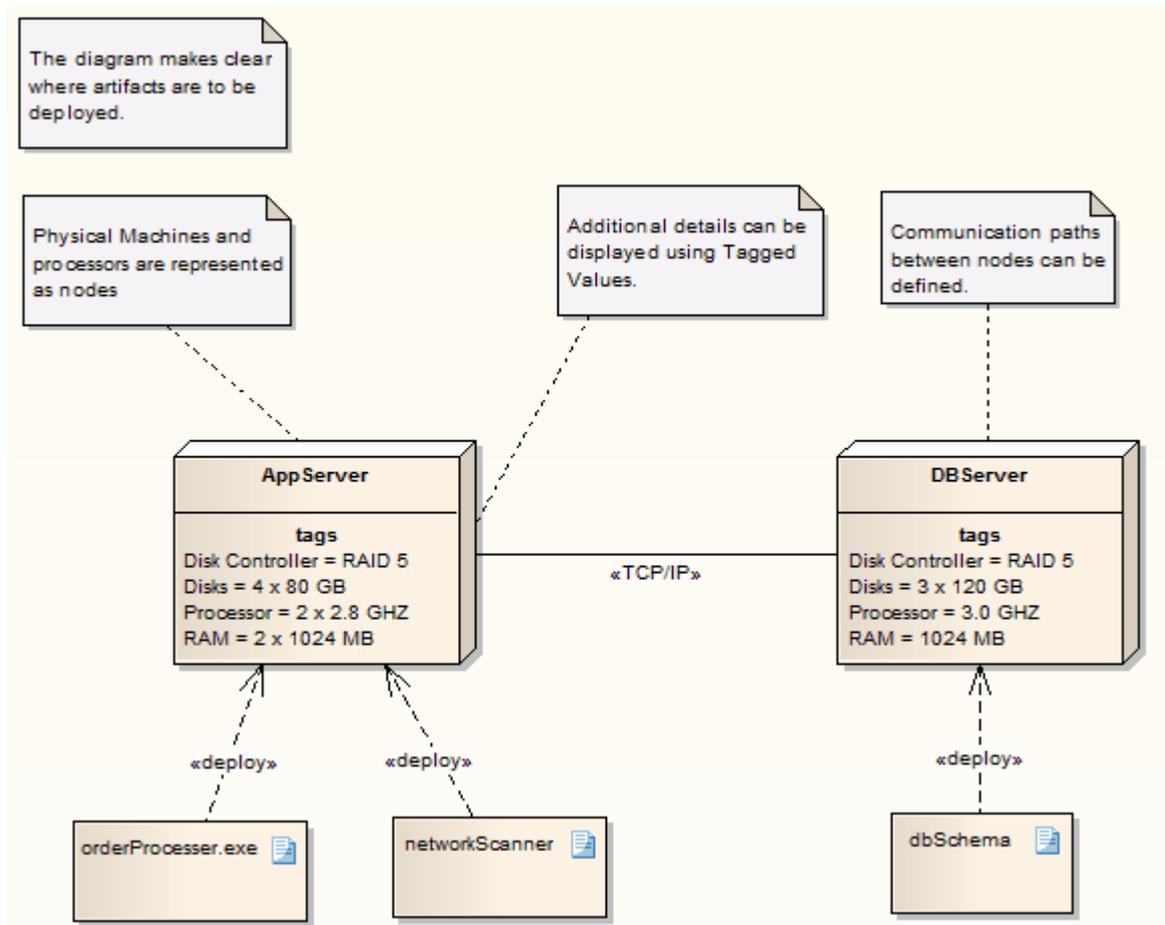
**Learn More:**

- [Components](#) ^[948]
- [Requirements Model](#) ^[521]
- [Component Diagram](#) ^[807]
- [The Component Model](#)

5.2.2.8 Deployment Model Template

The *Deployment model* describes how and where a system is to be deployed.

Physical machines and processors are represented by Nodes, and the internal construction can be depicted by embedding Nodes or Artifacts. As Artifacts are allocated to Nodes to model the system's deployment and roll out, the allocation is guided by the use of deployment specifications.

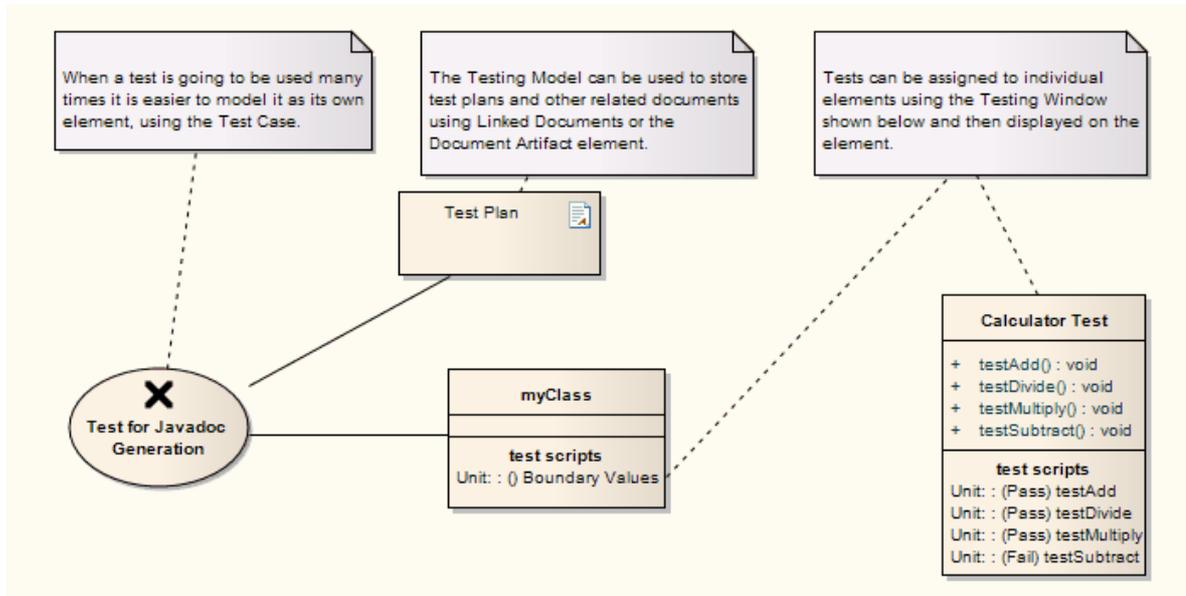


Learn More:

- [Nodes](#) ^[95]
- [Artifacts](#) ^[94]
- [Deployment and Roll Out](#) ^[65]
- [Deployment Diagram](#) ^[80]
- [Compartments - Tagged Values](#) ^[65]
- [The Physical Model](#)

5.2.2.9 Testing Model Template

The **Test model** describes and maintains a catalogue of tests, test plans and results that are executed against the current model.

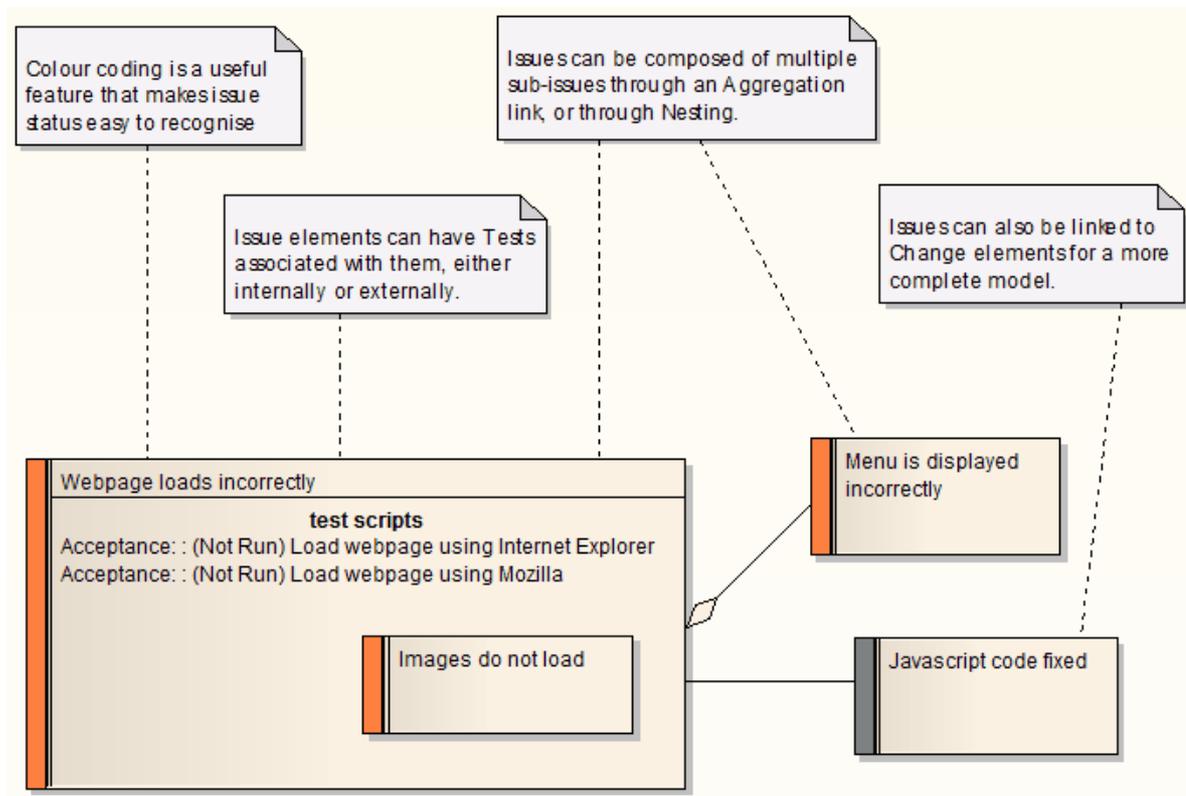


Learn More:

- [Testing](#)¹⁶⁹⁴
- [Test Case](#)¹⁶⁷⁴
- [Show Test Script Compartments](#)¹⁷⁰⁷
- [Testing Support in Enterprise Architect](#)

5.2.2.10 Maintenance Model Template

The *Maintenance model* enables visual representation of issues arising during and after development of a software product. The model can be enhanced with the integration of change elements and testing.



Learn More:

- [Development](#)^[1713]
- [Color Code External Requirements](#)^[1159]

5.2.2.11 Project Model Template

The *Project model* details the overall project plan, phases, milestones and resourcing requirements for the current project.

Project Managers can use Enterprise Architect to assign resources to elements, measure risk and effort and to estimate project size. Change control and maintenance are also supported.

The Project Model details the overall project plan, phases, milestones and resourcing requirements for the current project.

[Read About Project Management in Enterprise Architect](#)

Release Plan

Resources

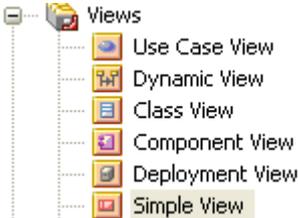
Learn More:

- [Project Management](#)³⁴⁶
- [Project Manager](#) (Online Resource)

5.3 Views

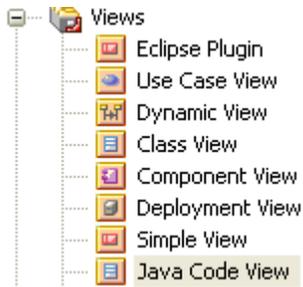
The top level packages in a model (below the project root nodes) can be created as *Views*.

Topics:

Topic	Detail	See also
Views	<p>Views are used simply to subdivide the model into partitions such as Business Process, Logical View or Dynamic View. Unlike Model Packages, they do not have automatically-generated components and can be created only under a root node. They are a good way to extend the model depending on specific requirements and modeling techniques.</p> 	
View Types	<p>There are six main types of View, each with their own package icon:</p> <ul style="list-style-type: none"> • Use Case View - containing, for example, Use Case diagrams, Analysis diagrams • Dynamic View - containing, for example, Activity diagrams, Communication diagrams, Sequence diagrams, State Machine diagrams • Class View - containing, for example, Class diagrams, Code Engineering, Data Models • Component View - containing, for example, Component diagrams • Deployment View - containing, for example, Deployment diagrams • Simple View - to customize your own type of view • You can use the first five categories, or devise your own based on the Simple View. You can create Views, rename them, move them into a different order, or delete them. Do this by right-clicking the mouse on the selected View to open the context menu, and choose the appropriate option 	<p>Use Case diagrams ^[813]</p> <p>Analysis diagrams ^[1187]</p> <p>Activity diagrams ^[811]</p> <p>Communication Diagrams ^[859]</p> <p>Sequence Diagrams ^[849]</p> <p>State Machine Diagrams ^[815]</p> <p>Class Diagrams ^[798]</p> <p>Code Engineering ^[1386]</p> <p>Data Models ^[1278]</p> <p>Component Diagrams ^[807]</p> <p>Deployment Diagrams ^[804]</p> <p>Add Additional Views ^[531]</p> <p>Rename Views ^[531]</p> <p>Delete Views ^[532]</p>

5.3.1 Add Views

The example below shows a customized view called *Java Code View*, which has been appended to the end of the Views list.



How to:

Step	Action	See Also
1	Right-click on the model root node in the Project Browser The context menu displays	
2	Select the New View menu option The Create New View dialog displays	
3	In the Name field, type the name of the View.	
4	In the Set View Icon Style panel, click on the radio button for the required View icon.	
5	If the model root node is under version control, the Add to Version Control checkbox displays, defaulted to selected If you do not want the new View to also be under version control, deselect the checkbox.	Version Control ^[242]
6	Click on the OK button.	

5.3.2 Rename Views

How to:

To rename a view, follow the steps below:

Step	Action	See Also
1	Right-click on the View in the Project Browser The context menu displays.	
2	Select the Properties menu option The Package Properties dialog displays	
3	In the Name field, type the new name and click on the OK button	

5.3.3 Delete Views

How to:

To delete a view, if necessary, follow the steps below

Step	Action	See Also
1	In the Project Browser , right-click on the view to delete The context menu displays	
2	Select the Delete <viewname> option A warning that child elements are to be deleted is displayed	
3	To delete the view and its contents, click on the Yes button To cancel the deletion, click on the No button	

Notes:

- If you delete a view, all its contents are deleted at the same time. It cannot be restored

5.4 Packages



A *package* is a container of model elements. It is represented in the **Project Browser** by the 'folder' icon familiar to Windows users.

This topic explores the tasks you can perform with packages, including:

- Open a package
- Add a package
- Rename a package
- Copy a package
- Drag a package onto a diagram
- Show or hide a package
- Delete a package
- List the elements in a package, with their properties, on the Package Browser.

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Element permission to update or delete a package

Learn More:

- [Open a package](#) ^[535]
- [Add a package](#) ^[534]
- [Rename a package](#) ^[534]
- [Copy a package](#) ^[535]
- [Drag a package onto a diagram](#) ^[536]
- [Show or hide a package](#) ^[537]
- [Delete a package](#) ^[537]
- [Package Browser](#) ^[456]
- [Update Element](#) ^[205]

5.4.1 Open Package in the Project Browser

To open a package from the **Project Browser**, follow the steps below:

How to:

Step	Instruction	See Also
1	Double-click on a package; the contents display in the Project Browser	
2	Click on the + and - symbols next to the folder icon to open or close the package respectively	

Notes:

- Package contents are arranged alphabetically and elements can be dragged from one package to another using the mouse

5.4.2 Add a Package

How to:

To add a package to the model hierarchy, follow the steps below:

Step	Instruction	See Also
1	In the Project Browser, select the package or view under which to add a new package.	
2	Right-click on the folder icon within the Project Browser. The context menu displays.	
3	Select the Add Add Package menu option. The New Model Package dialog displays.	
4	In the Package Name field type the name of the new package.	
5	To immediately create a diagram for the package, leave the Automatically add new diagram checkbox selected. To avoid creating a diagram, deselect the checkbox.	
6	If you are adding a package to a parent package that is under version control, the Add to Version Control option displays, with the checkbox selected. Deselect the checkbox to exclude the new package from version control, otherwise leave it selected.	Version Control ^[242]
7	Click on the OK button. The new package is inserted into the tree at the current location and, if you left the Automatically add new diagram checkbox selected, the New Diagram dialog displays.	New Diagrams ^[568]
8	If you have selected to put the package under version control, the Package Control Options dialog displays. Complete this dialog as required.	Configure Packages ^[329]

Notes:

- You can also add a new package element by dragging the Package icon from the Toolbox into a diagram. In this case the package is created under the diagram's owning package, and is created with a default diagram of the same type as that in which the package is created
- In a multi-user environment, other users do not see the change until they reload their project

Learn More:

- [Refresh View/Model](#) ^[187]

5.4.3 Rename a Package

How to:

To rename a package, follow the steps below.

Step	Instruction	See Also
1	Select the package to rename in the Project Browser	
2	Right-click to display the context menu	
3	Click on the Package Properties option	
4	In the Name field, type the new name	
5	Click on the OK button	

Notes:

- In a multi-user environment, other users do not see the change until they reload their project
- Alternatively, highlight the package to rename, and press **(F2)**

Learn More:

- [Refresh View/Model](#)^[187]

5.4.4 Copy a Package

Enterprise Architect enables you to quickly duplicate a complete package, including its child packages, elements and diagrams.

Topics:

Topic	Detail	See Also
Inserting a copy of a package	<p>You can insert the copy of a package:</p> <ul style="list-style-type: none"> • Under the same parent • Under one or more other packages in the same model or project, or • Under one or more other packages in any other model or project <p>This procedure is effectively the same as exporting and importing the package XMI file, with the Strip GUIDs checkbox selected. You would tend to use this procedure for copying sections of a model within the project rather than reproducing an entire model or project, although copying these larger structures is equally feasible.</p>	Package XMI files ^[344]
Copying a package	In the Project Browser, right-click on the required package and select the Copy Package to Clipboard context menu option (or click on the package and press (Ctrl+C)). The Copy Package to Clipboard dialog briefly displays until the copy operation completes.	
Pasting a package	<p>In the Project Browser, right-click on the package into which to paste the copied package, and select the Paste Package from Clipboard context menu option (or click on the package and press (Ctrl+V)). The Paste Package from Clipboard dialog briefly displays until the paste operation completes.</p> <p>The target package is expanded to expose the pasted package in the Project Browser. If you are pasting the package within the same model as the copied source, the source parent package</p>	

Topic	Detail	See Also
	<p>is also collapsed.</p> <p>If the target package already contains:</p> <ul style="list-style-type: none"> • a package with the same name as the pasted package, the pasted package name has the suffix - Copy • a package with the same name as the pasted package including the - Copy suffix, the suffix becomes - Copy1 (or - Copy2, - Copy3 and so on, as copies of the package accumulate in the target parent package) <p>You can keep the same package name as the source, or you can rename the package either by clicking twice on it and editing the name in the Project Browser, or by double-clicking on it and editing the name in the Properties dialog.</p>	

Notes:

- A copy of a package does not have the external cross references of the source package; that is, the following connectors are discarded:
 - Connectors coming from packages and elements outside the package being copied, into the package being copied
 - Connectors going to packages and elements outside the package being copied, from the package being copied
- You cannot paste a package into a parent package that is locked by another user or that is checked in. The **Paste...** option is grayed out in the context menu

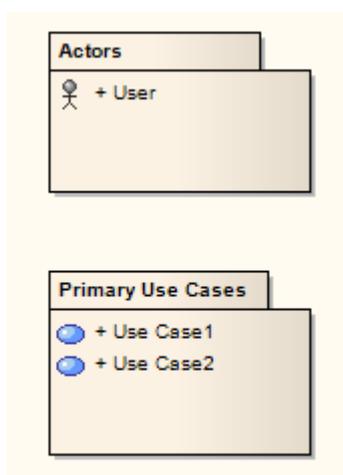
Learn More:

- [Locking Packages](#) ^[212]
- [Checking In/Checking Out Packages](#) ^[275]

5.4.5 Drag a Package Onto a Diagram

You can drag a package element from the **Project Browser** onto the current diagram, as an icon of the package listing its contents. This is a useful feature to help organize the display and documentation of models.

The following illustration shows how a package is displayed in a diagram; note the child Actor and Use Case icons.



Learn More:

- [Show or Hide Package Contents](#)^[537]

5.4.6 Show or Hide Package Contents

To show or hide the contents of packages in a diagram, follow the steps below:

How to:

Step	Instruction	See Also
1	Load a diagram	
2	Double-click in the background area to open the Diagram Properties dialog	
3	Click on the Elements tab	
4	Select or clear the Package Contents checkbox as required	
5	Click on the OK button	

5.4.7 Delete a Package**How to:**

To delete a package, follow the steps below:

Step	Instruction	See Also
1	Highlight the package in the Project Browser .	
2	Right-click to open the context menu.	
3	Click on the Delete option. A confirmation prompt displays.	
4	Click on the OK button.	

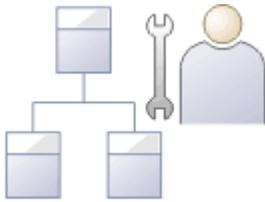
Notes:

- Deleting a package also deletes all contents of the package, including sub-packages and elements. Make very sure that you really want to do this before proceeding
- In a multi-user environment, other users do not see the change until they reload their project

Learn More:

- [Refresh View/Model](#)^[187]

5.5 Diagrams



Diagrams are collections of project elements laid out and inter-connected as required.

Enterprise Architect supports all of the UML diagrams, as well as several custom extensions. Together with the Enterprise Architect elements and connectors, these form the basis of the model. Diagrams are stored in packages and can have a parent object (optional). Diagrams can be moved from package to package.

Notes:

- If the diagram display is too small to read comfortably, click on the diagram, press and hold (**Ctrl**) and use the mouse wheel to temporarily expand or reduce the display magnification.

Learn More:

- [Diagram Context Menu](#) ^[538]
- [Diagram Tasks](#) ^[567]

5.5.1 Diagram Context Menu

To open the diagram context menu, open the required diagram and right-click on the diagram background.

Reference:

Option	Usage	Shortcut	See Also
Properties	Display the Diagram Properties dialog		Diagram Properties ^[569]
Scripts	List the scripts enabled for execution directly from the diagram (does not display if no diagram scripts exist)		Scripts ^[1819]
Swimlanes and Matrix	Add Swimlanes or a Swimlanes Matrix to the diagram		Swimlanes ^[590] Swimlanes Matrix ^[592]
Lock Diagram	Lock a diagram to protect it from inadvertent changes This does not apply if security is enabled in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, in which case you lock the model elements		Lock a Diagram ^[602] Lock Model Elements ^[210]
List View Gantt View	Display the diagram contents as a Diagram List or as a Gantt chart instead of as a diagram		Diagram List ^[462]
New Element or Connector	Insert various elements and connectors into a diagram		Insert Elements & Connectors ^[540]

Option	Usage	Shortcut	See Also
Insert Existing Element	Locate an existing element, using the Select Element dialog, and insert it into the diagram as a linked element		
Paste Element(s) as Link Paste Element(s) as New	Paste copied element(s) as a link or as new elements		Paste copied element ^[579]
Paste Image from Clipboard	Paste an image held on the clipboard into the diagram	Ctrl + Shift + Insert	Paste an image ^[75]
Make all elements selectable	<p>Make all the elements on the diagram selectable</p> <p>If an element is:</p> <ul style="list-style-type: none"> Selectable, you can move it around the diagram and perform right-click context-menu operations Unselectable, you cannot move it around the diagram and the only right-click operation available is to make the element selectable <p>This option has no effect on double-click operations on the element, such as displaying child diagrams</p>		
Modify Z Order	<p>Re-order overlapping objects in the diagram</p> <p>This option both:</p> <ul style="list-style-type: none"> Numbers the objects in the diagram by Z-order, in red, and Lists the objects by Z-order in the Modify Z Order dialog <p>In the dialog, you select the required object and move it up or down in the Z-order using the Up and Down arrow keys, each click on an arrow key changes the Z-order by 1</p> <p>When you select the object in the list, that object is also selected on the diagram and its Z-order number is shown in blue</p> <p>Click on the OK button to save your changes</p>		
Save as Profile	Save the current diagram as a Profile		Export a Profile ^[1050]
Import from source file(s)	Import, or reverse engineer, source code (not available in the Desktop edition)		Import, or reverse engineer, source code ^[1510]
Import DB schema from ODBC	Import database tables from an ODBC data source (not available in the Desktop edition)		Import database tables from an ODBC data source ^[1375]
Save Current Changes	Save any changes to the current diagram	Ctrl + S	
Help	View the Enterprise Architect Help on the type of diagram currently displayed		

Notes:

- Not all the menu options described appear on all diagram context menus

5.5.1.1 Insert Elements and Connectors

When you click on the **New Element or Connector** option on the **Diagram** context menu, a list of elements and connectors displays.

Topics:

Topic	Detail	See Also
Structure	<p>The structure of this list is as follows:</p> <ul style="list-style-type: none"> • Other - expands to offer options to select elements and connectors from diagram types other than either the current diagram type or pinned Toolbox pages • The expanded list of elements and connectors for the current diagram type • Collapsed lists of elements and connectors for pages that have been pinned in the Toolbox; if an MDG Technology: <ul style="list-style-type: none"> • is active • automatically pins Toolbox pages, and • has pages that redefine UML or Extended pages <p>The MDG Technology pages override the UML or Extended pages, which are not shown</p> • (At the end) Common - expands to display a list of the common elements and connectors 	

5.5.2 Print Preview

When you select the **File | Print Preview** menu option, the display initially shows the first two pages on one screen, with no scroll bar. To toggle between this two-page display and a single-page display, click on the  icon in the preview screen toolbar. In either mode, you can use the forward and back arrows to scroll through the pages of the diagram.

To display more than two pages on one screen, up to a maximum of ten pages, click on the **Zoom Out** button in the preview screen toolbar. The screen now includes the vertical scroll bar, which you can also use to scroll through the pages of the diagram.

Access: **File | Print Preview**

Use to:

- Preview pages before printing them

5.5.3 Diagram View

The Diagram View is the main workspace window that enables you to create and display diagrams.

Topics:

Topic	Detail	See Also
Opening a Diagram	<p>You open a diagram by double-clicking on the diagram name in the Project Browser; you can then open further diagrams in the same way</p> <p>You can also open a diagram from within an already open diagram by clicking on either hyperlinks or elements that contain other diagrams</p> <p>You can open many diagrams, but you can view only one at a time</p>	
Caption Bar	<p>Across the top of a diagram is the diagram caption bar, which includes the following:</p> <ul style="list-style-type: none"> • Icon and text label for the diagram type • Diagram name • Date and time the diagram was created • Date and time the diagram was last modified • The current magnification (zoom) of the diagram • The diagram page size, in pixels • A drop-down arrow that lists the currently-open diagrams; click on a diagram name to switch to that diagram, or to redisplay the hidden Start Page • The Window 'close cross'; click on this to close the displayed diagram 	
Format Toolbar	<p>Underneath the Caption Bar is the Format Toolbar, which enables you to apply a range of functions to the format and appearance of the elements in your diagram</p>	Format Toolbar ^[541]
Diagram View	<p>Use the Diagram View to build model relationships and elements</p> <p>Within the diagram, you can create new elements, drag in existing elements and generally organize the elements and relationships</p> <p>Most work on elements is carried out in the Diagram View, so understanding how it works and how to manipulate elements is essential; use the example project supplied with Enterprise Architect to explore the capabilities and behavior of the Diagram View</p>	

Notes:

- You can also use the Diagram List to manipulate elements

Learn More:

- [Diagram List](#)^[462]
- [Diagram Tabs](#)^[543]

5.5.3.1 Format Toolbar

The Format Tool toolbar can change the appearance of one or more selected elements by altering the font, color, line width and fill color. This can help to group, differentiate or make an element stand out.

An abbreviated Format Tool toolbar can also be displayed temporarily beside an element or connector. For elements, select one or more elements on the diagram, then click on the paintbrush icon beside the element. For connectors, right-click on the connector and the Format Tool toolbar appears above the context menu.

Use to:

- Modify the appearance of elements and connectors on a diagram

- Save a set of modifications as a user defined style
- Re-apply styles to other elements and connectors

Reference:



Action	Usage	Shortcut	See also
Set Font	Set the text font, style, size and effects for the selected element(s) using the Font dialog		Set Element Font ⁶⁵⁷
Text Color	Set the text color for the selected element(s) This option overrides the color setting of the Set Font option		
Fill Color	Set the fill color of the selected element(s)		
Line Color	Set the line color of the selected connector or element border		
Line Width	Set the line width of the selected connector or element border; choose a value between 1 (thinnest) and 5 (thickest)		
Apply Style	Apply a copied style to the selected elements or connector The style is copied using the Get Style option; if this is not used, the style defaults to the current setting of the Style List option		
Get Style	Copy the current style of the selected element or connector		
Style List	Apply a style from the saved style list to the selected element or connector These styles are saved to the list using the Save Style option		
Save Style	Click on the required element, click on the down arrow and select the appropriate option: <ul style="list-style-type: none"> • Save the style of the element to the Style List as a new style; enter the name of the style when prompted to do so • Save the (updated) style of the element under its current style name in the Style List • Refresh the Style List with any changes to the existing styles • Delete the style of the selected element from the Style List (refresh the list after selecting this option) 		
Align Left	Align the selected elements on the left border of the last-selected element	Ctrl + Alt + Left Arrow	
Align Right	Align the selected elements on the right border border of the last-s-selected element	Ctrl + Alt + Right Arrow	

Align Top	Align the selected elements on the top edge of the last-selected element	Ctrl + Alt + Up Arrow	
Align Bottom	Align the selected elements on the bottom edge of the last-selected element	Ctrl + Alt + Down Arrow	
Move to Top	Bring the selected element(s) to the top of the Z order		Diagram Context Menu [538]
Send To Back	Move the selected element(s) to the bottom of the Z order		Diagram Context Menu [538]
Diagram Auto-layout	Auto-layout the diagram (not for Behavioral diagrams) in one of the standard layouts, or modify a standard layout and apply that		Layout Diagrams [604]
Diagram Properties	Display the Diagram Properties dialog for the diagram	F5	Diagram Properties [569]
Delete	Delete the selected element(s)	Ctrl + D	
Add or Remove Buttons	This drop-down arrow at the right end of the toolbar enables you to: <ul style="list-style-type: none"> • Display or hide icons currently defined for the toolbar, or • Customize the toolbar to include other icons, to initiate commands found on other toolbars or totally new commands <p>(As an example, the last nine options on the Format toolbar are also found on the Diagram toolbar, which contains other options that could be duplicated here)</p>		The Customize Dialog [118] Diagram Toolbar [111]

Notes:

- To set the global appearance of all elements throughout a model, use the Options dialog; select the **Tools | Options** menu option, then select **Standard Colors** and **Diagram | Appearance** from the options tree
- To override the global appearance and define a default appearance of a selected element (or several selected elements) on all diagrams on which it occurs, set the Default Appearance for that element; right-click the element and choose the **Appearance | Default Appearance** context menu option

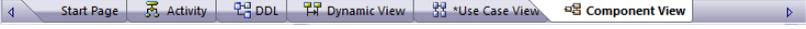
Learn More:

- [Standard Colors](#) [424]
- [Appearance](#) [424] (Diagram)
- [Set An Element's Default Appearance](#) [644]

5.5.4 Diagram Tabs

Diagram tabs are located, by default, at the bottom of the Diagram View. You can also move them to the top of the Diagram View, above the Status Bar.

Topics:

Topic	Detail	See Also
Access	<p>Each time you open a diagram, the diagram name and diagram type symbol are shown in the tab for easy identification and access</p>  <p>You can manipulate the tabs using the Diagram Tabs context menu (below) or the View Tab Switcher window</p>	View Tab Switcher ^[545]
The Diagram Tabs Menu	To access the Diagram Tabs context menu, right-click on an appropriate tab	

Reference:

Option	Usage	Shortcut	See also
Save Changes to '<tab name>'	Save the changes made to the diagram.		
Save All	Save the model.		
Reload '<tab name>'	<p>Reopen the diagram without the unsaved changes; that is, revert to the state before any changes were made.</p> <p>Refresh the diagram^[187] from the repository, to show any changes made by other users in a shared model.</p>		Refresh the diagram ^[187]
Close '<tab name>'	<p>Close the diagram; Enterprise Architect prompts you to save changes to the diagram.</p> <p>You can also close the tab by clicking the <i>middle</i> mouse button/scroller.</p>		
Close All	Close all open diagrams; Enterprise Architect prompts you to save any diagrams with unsaved changes.		
Close All Except '<tab name>'	Close all diagrams except for '<tab name>'; Enterprise Architect prompts you to save any diagrams with unsaved changes.		
Close All <view type>	<p>(Where several views of the same type can be opened at the same time, such as diagrams, RTF documents, or text editors.)</p> <p>Close all views of the same type as the selected tab, leaving views of other types still open.</p>		
Hide	(Start Page only.) Hide the Start Page . To redisplay it, use the drop-down menu at the top right of the Diagram View, or select the Window Show Start Page menu option.		

Notes:

- The **Component View** tab is white with bold text; this means that the Component View diagram is the

diagram currently displayed in the **Diagram View**

- The **Use Case View** tab has an asterisk. This means that there are unsaved changes in the Use Case View diagram
- There are also tabs for other types of view, such as the **Start Page** (as shown), RTF reports, code editors, the **Model Search** and the **Relationship Matrix**
- You can change the sequence of tabs by clicking on each tab and dragging it to a different position, as required. However, you cannot change the position of the **Start Page** tab. When displayed, it always sits at the left-hand end of the sequence

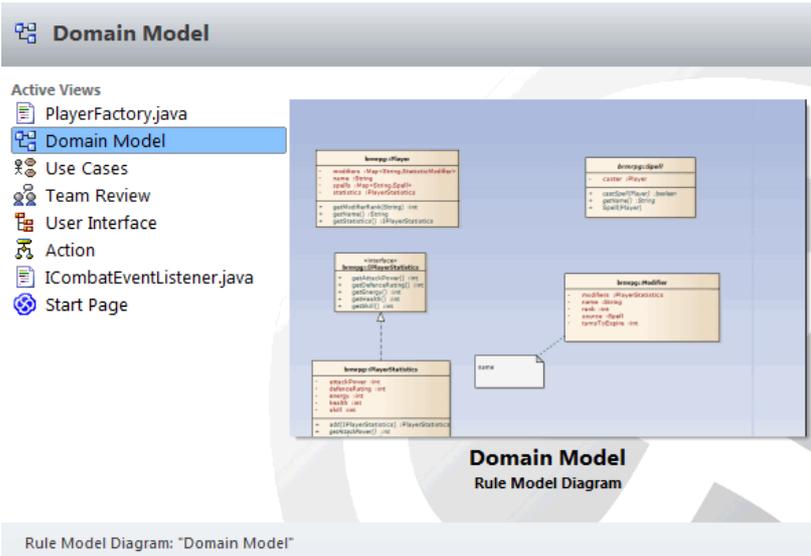
Learn More:

- [General Settings](#)^[422]

5.5.4.1 View Tab Switcher

The **View Tab Switcher** facilitates navigation between tabs of the Diagram View (such as **Diagrams**, **Code Editors** and the **Model Search** screen) that are currently open in Enterprise Architect.

Topics:

Topic	Detail	See Also
<p>Usage</p>	 <p>The screenshot shows the 'View Tab Switcher' window with a list of tabs on the left: PlayerFactory.java, Domain Model (selected), Use Cases, Team Review, User Interface, Action, ICombatEventListener.java, and Start Page. The main area displays a UML class diagram for the 'Domain Model' with classes like Player, Spell, PlayerStatistics, and Modifier.</p> <p>To display the View Tab Switcher press and hold (Ctrl) and then press (Tab). The View Tab Switcher lists the currently opened tabs, in order of most recently activated to least recently activated. The selection cursor is on the tab that was activated just before the currently active one. Details of the selected tab are displayed on the right hand side of the screen, and include:</p> <ul style="list-style-type: none"> • A preview thumbnail of the tab, and • A description of the tab and its contents <p>Continue holding the (Ctrl) key to keep the window open. To move the selection cursor down the list of tabs press the (Tab) key; to move the selection cursor up the list, press the (Shift+Tab) keys. Alternatively, use the arrow keys.</p>	
<p>Activating a tab</p>	<p>To activate a tab in the Diagram View, either release the (Ctrl) key or press the (Enter) key. Alternatively, click the left mouse button on the</p>	

Topic	Detail	See Also
	required tab. Activating a tab moves it to the top of the Active Views list, whereas closing a tab removes it from the list completely.	

5.5.5 Diagram Toolbox

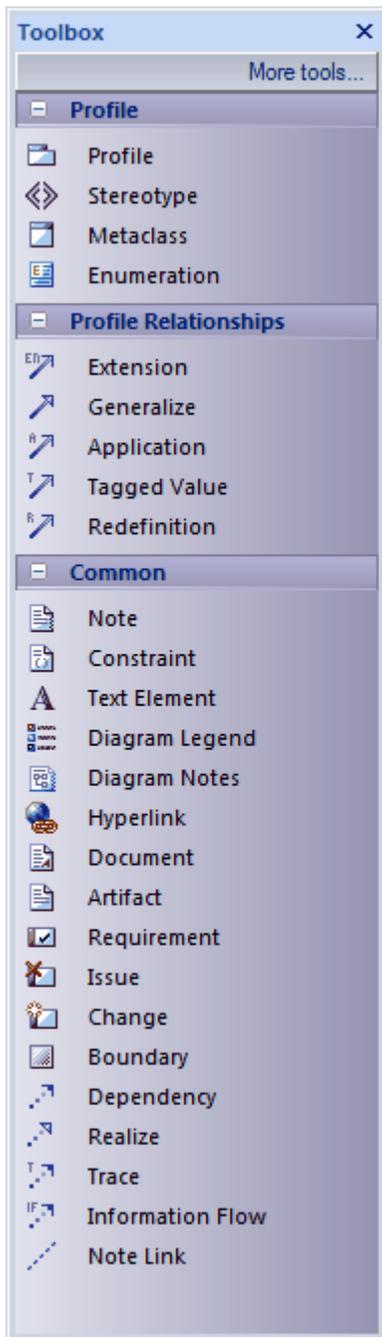
The Enterprise Architect Toolbox is a panel of icons that you use to create elements and connectors on a diagram.

Within the Toolbox, related elements and connectors are organized into pages, each page containing the elements or connectors used for a particular type of diagram. The diagrams include standard UML diagrams, Enterprise Architect Extended diagrams, and any MDG Technologies and Profiles that you have added to Enterprise Architect.

When you open a diagram, the Toolbox automatically provides the element and relationship pages corresponding to the diagram type. This does not prevent you using elements and connectors from other pages in a given diagram, though some combinations might not represent valid UML.

Access: [View | Toolbox \(Alt+5\)](#)

Example:



Topics:

Topic	Detail	See Also
<p>Displaying specific pages</p>	<p>To display specific pages in the Toolbox, select the More Tools option at the top of the Toolbox and select the appropriate UML, Extended or customized diagram type from the menu</p> <p>In most cases, three pages display: <type> elements, <type> Relationships and Common; if you select the <default> option, you display only the Common page</p>	

Topic	Detail	See Also
Customizing the Toolbox	You can customize the Toolbox pages by pinning them within the Toolbox, or by adding MDG Technologies and UML Profiles to the Toolbox	Toolbox Appearance Options ^[549] MDG Technologies ^[1033] UML Profiles ^[1028]

How to Create Elements and Connectors:

Step	Instruction	See Also
1	In the Project Browser, double-click on the icon against the required diagram The diagram opens with the appropriate Toolbox pages for that diagram type If you want a different set of elements and connectors, click on More tools and select the appropriate diagram type as explained above	
2	Click on the required item; for example, the Class element or Associate relationship	
3	For element items, click anywhere on the diagram to place the new element	
4	For connector items, drag the cursor between the source and target elements on the diagram; the solid border of the elements changes to a hatched border as you pass the cursor over them, indicating the source and potential target elements To add bends to the connector, press (Shift) as you change the drag direction of the cursor Alternatively, drag from the source element to an empty area of the diagram; the Quick-linker enables you to create the target element	Quick-linker ^[623]
5	Edit the element properties or connector properties, as required	Properties Dialog ^[660] Connector Properties ^[756]

Notes:

- Dropping a Package element from the Toolbox into a diagram creates a new package in the Project Browser, and a default diagram of the same type as the current diagram
- If you are creating several elements of one type, after creating the first just press (**Shift+F3**) or (**Ctrl**) + **click** to create the next element of that type; for connectors, click on the source element and press (**F3**) to create another connector of the same type
- You can change an unstereotyped element to one of its stereotyped elements by dragging the stereotyped element from the Toolbox onto the unstereotyped element in the diagram; for example, you can stereotype a Class by dragging a Table element or a Profiled Class element onto it
- As you drag the stereotyped element, you can press (**Ctrl**) to apply the element stereotype to the unstereotyped element, or (**Shift**) to add the stereotyped element as a new element; otherwise a small context menu displays offering these actions as options
- If the diagram element already has the stereotype, you can also drag the Toolbox element onto it to synchronize the element's stereotype Tagged Values
- The Toolbox can be docked on either side of the diagram, or free floated on top of the diagram to expose more surface for editing

- You can also hide and show the whole Toolbox using the  **Show Element Toolbox** button on the Workspace Views toolbar
- Enterprise Architect provides Toolbox pages for a wide range of MDG Technologies, such as ArchiMate, BPEL, BPMN, Data Flow Diagrams, ICONIX and Mind Mapping, as part of the initial install

Learn More:

- [Stereotype Dialog](#) ^[1016]
- [Synchronize Tags and Constraints](#) ^[1029]
- [ArchiMate](#) ^[1272]
- [BPEL](#) ^[1235]
- [BPMN](#) ^[1219]
- [Data Flow Diagrams](#) ^[1184]
- [ICONIX](#) ^[1542]
- [Mind Mapping](#) ^[1182]

5.5.5.1 Toolbox Appearance Options

You can modify the appearance of the Toolbox pages in several ways, through the context menu.

Access: **Right-Click on the Toolbox page**

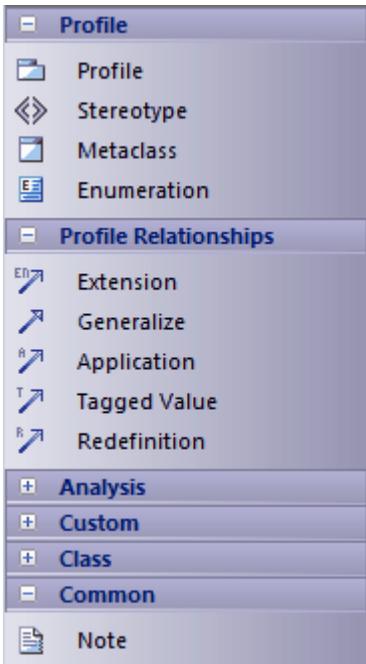
Topics:

Topic	Detail	See Also
Hiding labels	<p>To hide the element or relationship labels (and subsequently redisplay them), select the Hide Labels (or Show Labels) context menu option. The icons in the page then 'wrap' within the page, without text labels.</p>  <p>When you hide the labels, you can display the label of an individual element or relationship by moving the cursor over the icon.</p>	
Pinning pages	<p>To 'pin' the page so that it displays for any group in the Toolbox, select the Pin in Toolbox menu option. (This is not available on the Common page, which displays in all groups anyway.)</p> <p>For example, if you 'pinned' the Class elements page, and switched to the Communication pages, the Toolbox would include a collapsed Class elements page underneath the Communication pages.</p>	
Unpinning pages	<p>To unpin the page so that it displays only in its own Toolbox group, right-click on it and select the Unpin from Toolbox context menu option.</p>	
Collapsing pages	<p>To collapse a page to just show the heading (<type> elements, <type> Relationships or Common), click on the 'minus' box at the left of the page heading. To expand the page again, click on the heading. Alternatively, collapse the page by right-clicking on the page and</p>	

Topic	Detail	See Also
	selecting the Collapse context menu option.	

How to:

To tailor the Toolbox to list all pages you require at the same time, follow the steps below:

Step	Instruction	See Also
1	Click on the Set Toolbox Visibility context menu option The Visible Toolbox Pages dialog displays	
2	By default, the dialog lists the element pages only, in the order: UML pages, Extended pages, MDG Technology pages To include the corresponding relationship pages, select the Show Relationship Pages checkbox at the bottom of the dialog.	
3	For each page to display on the Toolbox, select the Visible checkbox Deselect the checkbox if you no longer require a page to be displayed	
4	When you have defined the list of pages to display, click on the OK button The pages you have selected are pinned to the Toolbox in a collapsed state, underneath the current diagram-type pages 	
5	To expand a page, click on the heading You can remove a page individually by expanding it, right-clicking on it and selecting the Unpin from Toolbox context menu option.	

Notes:

- On a Toolbox page for an MDG Technology or UML Profile, if you right-click directly on a profile element an additional option - Synchronize Stereotype - is available at the end of the context menu. This enables you to synchronize the Tagged Values and constraints for all elements created from the selected profile element
- MDG Technologies can impose their own Toolbox page visibility. For example, if ICONIX is the active technology, all six ICONIX pages are automatically exposed in the Toolbox. If the active Technology pages duplicate UML or Extended pages (as the ICONIX pages do) then the pinned Technology pages override and replace the pinned UML and Extended pages. For example, if ICONIX is active and you have pinned the Extended Analysis page, the Analysis page in the list is the ICONIX-defined page, not the Extended Analysis page

Learn More:

- [Synchronize Tags and Constraints](#) 

5.5.5.2 Toolbox Shortcut Menu

To add elements and connectors into a diagram, you can access the **Toolbox** shortcut menu instead of employing the full graphical Toolbox.

Topics:

Topic	Detail	See Also
Usage	<p>The menu provides options to select:</p> <ul style="list-style-type: none"> • Elements specific to the current diagram type • Relationships specific to the current diagram type • Elements and relationships from any pages pinned in the Toolbox • Common elements and relationships • Elements and connectors for other diagram types <p>The advantage of using the Toolbox shortcut menu is that it provides an increased amount of the workspace to be used for diagramming rather than to display fixed (instead of pop-up) menus</p>	

To use the Toolbox shortcut menu, follow the steps below:

Step	Action	See Also
1	Open a diagram	
2	<p>Either:</p> <ul style="list-style-type: none"> • Click on the diagram background and press (Insert) or (Spacebar) • Press and hold (Ctrl) and right-click on the diagram background <p>The shortcut menu displays, listing the current diagram-type elements and connectors</p>	
3	If necessary, select the Other option or a pinned Toolbox page option to list elements and connectors for a different diagram type	
4	Select the element or connector to include in the diagram. The object is added to the diagram.	

Step	Action	See Also
	If you select the Other context menu option, the final menu in the sequence offers the Activate <Type> Toolbox option; this opens and activates the corresponding page in the Toolbox , if the Toolbox is visible	

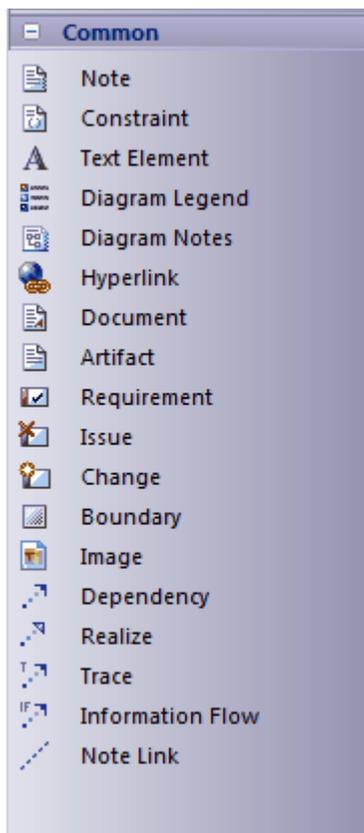
Notes:

- As with the Toolbox itself, if an MDG Technology:
 - Is active
 - Automatically pins Toolbox pages, and
 - Duplicates UML or Extended pages

The pinned Technology pages override and replace the pinned UML or Extended pages in the initial Toolbox shortcut menu

5.5.5.3 Common Group

The **Common** page of elements and relationships is displayed at the bottom of every other set of pages. It contains the elements and relationships that can be used on any diagram.

**Toolbox Elements and Connectors:**

Click on the elements and connectors below for information.

Common Elements	Common Connectors
 Note	 Dependency
 Constraint	 Realize
 Text Element	 Trace
 Diagram Legend	 Information Flow
 Diagram Notes	 Note Link
 Hyperlink	
 Document	
 Artifact	
 Requirement	
 Issue	
 Change	
 Boundary	
 Image	

5.5.5.4 Use Case Group

Use Case elements are used to build Use Case models. These describe the functionality of the system to be built, the requirements, the constraints and how the user interacts with the system.

Often, Sequence diagrams are associated with Use Cases to capture work flow and system behavior.

Topics:

Image	Topic	Detail	See Also
 Use Case  Actor  Use Case  Test Case  Collaboration  Collaboration Use  Boundary  Package  Use Case Relationships  Use  Associate  Generalize  Include  Extend  Realize  Invokes  Precedes	Usage	<p>The Use Case group is used to model the system functionality from the perspective of a system user; the user is called an Actor and is drawn as a stick figure, although the user could be another computer system or similar</p> <p>A Use Case is a discrete piece of functionality the system provides that enables the user to perform some piece of work or something of value using the system</p> <p>Examples of Use Cases are: login, open account, transfer funds, check balance and logout; each of these implies some purposeful and discrete functionality the system is to provide to a user</p> <p>A Test Case describes what must be set up in order to test a</p>	Test Case <small>[1674]</small>

Image	Topic	Detail	See Also
		<p>particular feature</p> <p>The connectors available include: associate (an actor uses a Use Case), extend (one Use Case can extend another), include (one Use Case can include another) and realize (this Use Case might realize some business requirement)</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram; set an element name and other properties as prompted</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element</p>	

Notes:

- Invokes and Precedes relationships are defined by the Open Modeling Language (OML). They are stereotyped Dependency relationships; *Invokes* indicates that Use Case A, at some point, causes Use Case B to happen, whilst *Precedes* indicates that Use Case C must complete before Use Case D can begin

Learn More:

- [Use Case Models](#) ⁸¹³

5.5.5.5 Class Group

The **Class** group can be used for Package diagrams, Class diagrams and Object diagrams: those that usually display elements concerned with the logical structure of the system, such as Objects, Classes and Interfaces.

Logical models can include domain models (high level business driven object model) through to strict development Class models (define inheritance, attributes, operations).

Topics:

Image	Topic	Detail	See Also
	Usage	<p>The Class group is used for creating Class models and database models. Class modeling is done using the Class and Interface elements, as well as occasional use of the Object element to model Class instances. You can add Association or Aggregation relationships. See the Class model template for an example of this.</p> <p>Use the Table element to insert a stereotyped Class for use in database modeling. See the Data Modeling topic for more details.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, click on the start element in the diagram and drag to the end element.</p>	<p>Class ⁵²³</p> <p>Data Modeling ¹²⁷⁶</p>

Learn More:

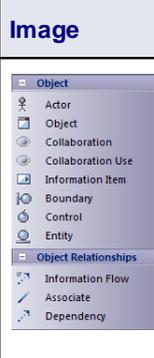
- [Package Diagrams](#) ^[796]
- [Class Diagrams](#) ^[798]
- [Object Diagrams](#) ^[799]

5.5.5.6 Object Group

The **Object** group is used to create Object diagrams.

Object diagrams reflect multiplicity and the roles instantiated Classes could serve. They are useful in creating different cases in which relationships and Classes are applied.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>The user is called an Actor and is drawn as a stick figure, although the user could be another computer system or similar.</p> <p>An Object is an instance of a Class.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Actor ^[877]</p> <p>Object ^[957]</p>

Learn More:

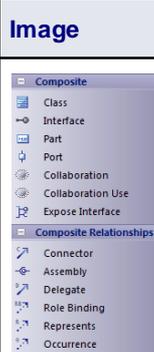
- [Object Diagrams](#) ^[799]

5.5.5.7 Composite Group

The **Composite** group is used for Composite Structure diagrams.

These reflect the internal collaboration of Classes, Interfaces or Components to describe a functionality or to express run-time architectures, usage patterns and the participating elements' relationships, which static diagrams might not show.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram; set an element name and other properties as prompted</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element</p>	

Learn More:

- [Composite Structure Diagrams](#)^[80]

5.5.5.8 Communication Group

The **Communication group** is used to model dynamic interactions between elements at run-time.

The Actor element models a user of the system, while the other elements model things within the system, including standard elements (rectangular element), user interface component (circle with left positioned vertical bar), controller (circle with arrow head in top most position) and entity (circle with bar at bottom).

Topics:

Image	Topic	Detail	See Also
	Usage	<p>Communication diagrams are used to model work flow and sequential passing of messages between elements in real time. They are often placed beneath Use Case elements to further expand on Use Case behavior over time.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	Communication Diagrams ^[85]

Notes:

- Communication diagrams were known as Collaboration diagrams in UML 1.4

5.5.5.9 Interaction Group

The **Interaction group** is used for Interaction diagrams (Sequence, Timing, Communication or Interaction Overview), which are used to model work flow and sequential passing of messages between elements in real time.

Interaction diagrams are often placed beneath Use Case elements to further expand on Use Case behavior over time.

Topics:

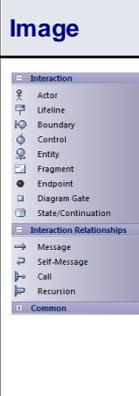
Image	Topic	Detail	See Also
	Usage	<p>The Interaction group is used to model dynamic interactions between elements at run-time</p> <p>The Actor element models a user of the system, while the other elements model things within the system, including standard elements (Lifeline), user interface component (Boundary), controller and Entity</p> <p>The meaning of the element symbols is discussed further in the Sequence diagram topic. The Message (sequence) relationship is used to model the flow of information and processing between elements.</p>	Sequence diagram ^[84]

Image	Topic	Detail	See Also
		<p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

Notes:

- Messages can be simple or recursive calls

Learn More:

- [Sequence](#) ^[849]
- [Timing](#) ^[830]
- [Communication](#) ^[859]
- [Interaction Overview](#) ^[861]

5.5.5.10 Timing Group

The **Timing** group is used solely for Timing diagrams, which use a time-scale to define the behavior of objects. The time-scale visualizes how the objects change state and interact over time.

Timing diagrams can be used for defining hardware-driven or embedded software components, and time-driven business processes.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>A Lifeline is the path an object takes across a measure of time, indicated by the x-axis.</p> <p>A State Lifeline follows discrete transitions between states, which are defined along the y-axis of the timeline. Any transition has optional attributes of timing constraints, duration constraints and observations.</p> <p>A Value Lifeline shows the lifeline's state across the diagram, within parallel lines indicating a steady state. A cross between the lines indicates a transition or change in state.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>State Lifeline ^[923]</p> <p>Value Lifeline ^[938]</p>

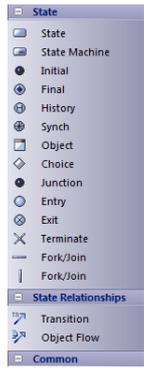
Learn More:

- [Timing Diagrams](#) ^[830]

5.5.5.11 State Group

The **State** group is used by State Machine diagrams to show the enableable states a Class or element might be in and the transitions from one state to another. These diagrams are often placed under a Class element in the Project Browser to illustrate how a particular element changes over time.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>The State group provides elements common to State Machine diagrams; basically the State, start and end nodes and the Object Flow relation. State Machine diagrams are used to model the states or conditions that elements might be in at runtime, such as active, inactive, idle, accelerating or braking. States can have substates; for example, accelerate or brake might be substates of active.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

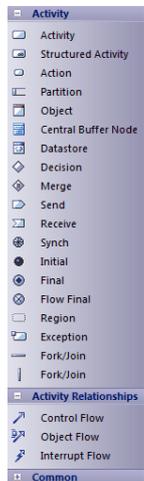
Learn More:

- [State Machine Diagrams](#) ⁸¹⁵

5.5.5.12 Activity Group

The **Activity** group is used to model system dynamics from a number of viewpoints in Activity diagrams and Interaction Overview diagrams.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>Activity elements enable you to describe the dynamics of the system from the point of view of activities and flows between them. Activities can be stereotyped as a process to display a business process icon. An Activity is some work that is carried out; it might overlap several Use Cases or form only a part of one Use Case.</p> <p>Send and Receive events are included as triggers.</p> <p>A Decision element marks a point where processing might split based on some outcome or value.</p> <p>The Flow relation models an active transition and synch points are used to split and rejoin periods of parallel processing.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

Learn More:

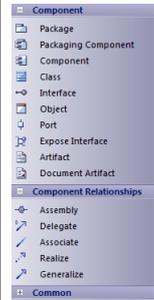
- [Activity Diagrams](#) ⁸¹
- [Interaction Overview Diagrams](#) ⁸⁶

5.5.5.13 Component Group

The **Component** group enables you to model the physical components of your system in a Component diagram.

A component is a piece of hardware or software that makes up the system; for example, a DLL or Web Server are Components that might be deployed on a Windows 2000 Server (Node). See the Deployment Diagram topic for an example of this.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>The Component group contains elements related to the actual building of the system: the components that make up the system (such as ActiveX DLL's or Java beans), the Interfaces they expose and the dependencies between those elements.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

Learn More:

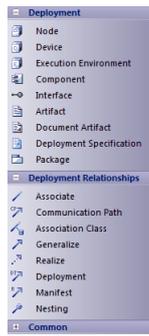
- [Component Diagram](#) ⁸⁰
- [Deployment Diagram](#) ⁸⁰

5.5.5.14 Deployment Group

The **Deployment** group enables you to model the physical components and deployment structure of your system in a Deployment diagram.

A Component is a piece of hardware or software that makes up the system, and a Node is a physical platform on which the component is to exist. For example, DLLs or Web Servers are Components that could be deployed on a Windows 2000 Server (Node). See the Deployment Diagram topic for an example of this.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>The Deployment group contains elements related to the actual building of the system; the components that make up the system (such as ActiveX DLLs or Java beans) and the nodes those components run on, including the physical connections between nodes.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

Learn More:

- [Deployment Diagram](#) 804

5.5.5.15 Profile Group

The **Profile** group contains extended UML elements and connectors that can be used to create and modify Profiles, for rapidly creating stereotypes and Tagged Values that can be applied to structures such as elements, attributes, methods and connectors.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>A Profile is used to provide a generic extension mechanism for building UML models in particular domains. They are based on additional Stereotypes and Tagged Values that are applied to structures such as elements, attributes, methods, connectors and connector ends.</p> <p>A Stereotype provides a mechanism for varying the behavior and type of a model element.</p> <p>A Metaclass is used to create a Class whose instances are Classes; a metaclass is typically used to construct metamodels.</p> <p>An Enumeration creates a Class stereotyped as enumeration, which is used to provide a list of named values as the range of a particular type.</p> <p>An Extension relationship shows that a stereotype extends one or more metaclasses. All stereotypes must extend either one or more Metaclasses, or another stereotype that extends a stereotype (that itself extends a stereotype, and so on).</p> <p>A Generalize relationship shows that one stereotype specializes a more general stereotype. The more general stereotype must still extend a metaclass.</p> <p>A Tagged Value relationship defines a reference-type (that is, RefGUID) Tagged Value owned by the source stereotype. The Tagged Value is named for the target role of this association, and is limited to referencing elements with the stereotype by the association target element.</p> <p>The Application and Redefinition relationships are deprecated.</p>	

Image	Topic	Detail	See Also
		<p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

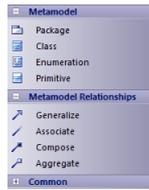
Learn More:

- [Create and Modify Profiles](#) ^[1026]

5.5.5.16 Metamodel Group

The **Metamodel** group enables you to create metamodel diagrams with support for **MOF** diagrams.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>A Package is a namespace as well as an element that can be contained in other package's namespaces.</p> <p>A Class is a representation of objects, that reflects their structure and behavior within the system.</p> <p>An Enumeration is a Class with an enumeration stereotype.</p> <p>A Primitive supports the MOF specification (deprecated - use the UML Primitive in the Class group).</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package ^[960]</p> <p>Class ^[941]</p> <p>Enumeration ^[952]</p> <p>Class Group ^[554]</p>

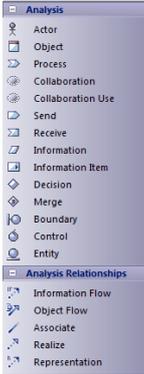
Learn More:

- [MOF](#) ^[1630]

5.5.5.17 Analysis Group

Analysis-type elements are used early in modeling to capture business processes, activities and general domain information. They are generally used in Analysis diagrams.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>The elements and relationships in the Analysis group are used for early modeling of business processes, activities and collaborations. You can use stereotyped activities to model business processes, or stereotyped elements to capture standard UML business process modeling extensions such as worker, case worker, entity, and controller.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, click on the start element in the diagram and drag to the end element.</p>	

Learn More:

- [Analysis Diagrams](#) ^[1187]

5.5.5.18 Custom Group

The **Custom** group contains a few extended UML elements that might be of use in modeling or designing your system in a Custom diagram.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>A Package is a namespace as well as an element that can be contained in other package's namespaces.</p> <p>A Requirement is a custom element used to capture requirements outside of standard UML elements. A Requirement expresses required system behavior that can cross several Use Cases. You can connect Requirements to other elements using the Realize connector to express the implementation of a requirement and hence the traceability from user requirements to what is being built.</p> <p>An Issue element is a structured comment that contains information about defects and issues relating to the system/model (see the Defects (Issues) topic). Affected elements are connected by Trace connectors.</p> <p>A Change element is a structured comment that contains information about changes requested to the system/model (see the Changes and Defects topic). Affected elements are connected by Trace connectors.</p> <p>A Screen provides a stereotyped Class element that displays a GUI type screen; this can be used to express application GUI elements and flows between them.</p> <p>A UI control likewise can be used to express GUI controls.</p> <p>A Test Case element defines what must be set up in order to test a particular feature (see the Testing Workspace topic). It enables you to define a set of tests once for a number of elements, and provides greater visibility for</p>	<p>Package ^[960]</p> <p>Requirements ^[1154]</p> <p>Traceability ^[493]</p> <p>Defects (Issues) ^[1719]</p> <p>Trace ^[1012]</p> <p>Changes and Defects ^[1719]</p> <p>Screen ^[1287]</p> <p>UI control ^[1282]</p> <p>Test Case ^[1674]</p> <p>Testing Workspace ^[1695]</p> <p>Table ^[1277]</p>

Image	Topic	Detail	See Also
		<p>tests.</p> <p>An Entity is a stereotyped element that represents any general thing not captured by the element or Class type elements (for example a trading partner). Use of this element is deprecated: it was originally intended to take the role now occupied by a Table element.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

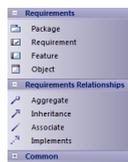
Learn More:

- [Custom Diagram](#) ^[1183]

5.5.5.19 Requirement Group

As an analysis step, often it is desirable to capture simple system *requirements*. These are eventually realized by Use Cases.

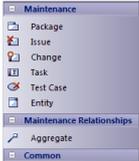
Topics:

Image	Topic	Detail	See Also
	Usage	<p>A Package is a namespace as well as an element that can be contained in other package's namespaces.</p> <p>Specify the Requirement of a system. Note that there are a few different requirement types, such as.</p> <ul style="list-style-type: none"> • Display • Functional • Performance • Printing • Report • Testing • Validate <p>A Feature is a small client-valued function expressed as a requirement. Features are the primary requirements-gathering artifact of the Feature-Driven Design (FDD) methodology.</p> <p>An Object is an instance of a Class.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package ^[960]</p> <p>Requirement ^[1154]</p> <p>Feature ^[1166]</p> <p>Object ^[957]</p>

5.5.5.20 Maintenance Group

The **Maintenance** elements are defects, changes, issues and tasks.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>A Package is a namespace as well as an element that can be contained in other package's namespaces.</p> <p>An Issue element is a structured comment that contains information about defects and issues relating to the system/model. Affected elements are connected by Trace connectors.</p> <p>A Change element is a structured comment that contains information about changes requested to the system/model. Affected elements are connected by Trace connectors.</p> <p>A Task element defines a task attached to an element, that enables resources to be assigned specifically to the task rather than just to the parent element. You can monitor these tasks on diagram GANTT charts.</p> <p>A Test Case describes what must be set up in order to test a particular feature.</p> <p>An Entity is a stereotyped element that represents any general thing not captured by the element or Class type elements (for example a trading partner). Use of this element is deprecated: it was originally intended to take the role now occupied by a Table element.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package ^[960]</p> <p>Defects and Issues ^[1719]</p> <p>Trace ^[1012]</p> <p>Change ^[1719]</p> <p>Task ^[1299]</p> <p>Test Case ^[1674]</p> <p>Table ^[1277]</p>

Learn More:

- [Maintenance](#) ^[1719]

5.5.5.21 User Interface Group

The **User Interface** group enables you to create graphical user interface diagrams.

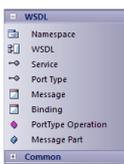
Topics:

Image	Topic	Detail	See Also
	<p>Usage</p>	<p>A Package is a namespace as well as an element that can be contained in other packages' namespaces.</p> <p>A Screen element represents a graphical user interface. You can place GUI elements onto the screen element.</p> <p>UI Control elements are placed onto the screen element to build up a graphical user interface diagram. There are different stereotyped elements such as buttons and combo boxes.</p> <p>An Object is an instance of a Class.</p> <p>To add an element to the current diagram, click on the required icon, and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	<p>Package^[960]</p> <p>Screen^[1281]</p> <p>UI Control^[1282]</p> <p>Object^[957]</p>
	<p>GUI Elements</p>	<p>The diagram on the left illustrates the GUI elements from the Toolbox, within a Screen element.</p>	

5.5.5.22 WSDL Group

The **WSDL** group gives you the ability to rapidly model and automatically generate W3C Web Service Definition Language (WSDL) documents.

Topics:

Image	Topic	Detail	See Also
	<p>Usage</p>	<p>A Namespace represents the top-level container for the WSDL model. Drag this element onto an open diagram to create the necessary model structure for WSDL documents.</p> <p>A physical WSDL document is represented as a UML component. Its interfaces represent the WSDL services.</p> <p>A WSDL Port Type is modeled as a UML interface. Its Port Type Operations are realized by Binding elements. Each of the operation parameters is derived from the Message elements defined in the Messages package.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, click on the start element in the diagram and drag to the end element.</p>	<p>WSDL Namespace^[1612]</p> <p>WSDL document^[1620]</p> <p>WSDL services^[1619]</p> <p>WSDL Port Type^[1614]</p> <p>Port Type Operations^[1615]</p> <p>WSDL Binding^[1617]</p> <p>WSDL Messages^[1613]</p>

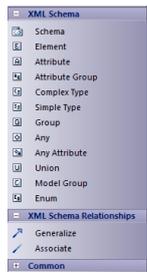
Learn More:

- [Model WSDL](#)^[1612]
- [Generate WSDL](#)^[1621]

5.5.5.23 XML Schema Group

The **XML Schema** group provides the ability to model and automatically generate W3C XSD schema files. This group implements the constructs provided by the UML profile for XML Schema.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>A Schema corresponds to a UML package, which contains the type and element definitions for a particular targetNamespace. Drag this item onto an open diagram to create the package to contain your schema model elements. The package is stereotyped as XSDschema.</p> <p>Open the logical diagram created under the XSDschema package and add additional schema elements as required.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set the name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, then click on the start element in the diagram and drag to the end element.</p>	

Learn More:

- [Model XSD](#) ^[1581]
- [Generate XSD](#) ^[1598]
- [UML profile for XML Schema](#) ^[1602]

5.5.5.24 Data Modeling Group

This group is used for database modeling and database design, in conjunction with the *UML Data Modeling Profile*.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>The Table element defines a table on the data model.</p> <p>The View element represents database views in the data model.</p> <p>The Procedure element represents stored procedures in the data model.</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p> <p>To add a relationship, click on the required icon, click on the start element in the diagram and drag to the end element.</p>	<p>Table ^[1277]</p> <p>Views ^[1370]</p> <p>Stored Procedures ^[1366]</p>

Learn More:

- [Database Modeling](#) ^[1276]

5.5.5.25 Test Domain Group

The Test Domain group is used to create Test Domain diagrams, used in the Testpoint facility. A Test Domain diagram provides specific elements that can aid in the logical composition of tests into Test Sets and Test Suites.

Topics:

Image	Topic	Detail	See Also
	Usage	<p>A <i>Test Cut</i> is a group of operations that represent a specific behavior (such as 'Print') defined for a Class, separate from other constraints defined for the Class</p> <p>A <i>Test Set</i> aggregates one or more groups of methods (Test Cuts) that perhaps span multiple Classes, into a single task</p> <p>A <i>Test Suite</i> aggregates one or more groups of tasks (Test Sets)</p> <p>To add an element to the current diagram, click on the required icon and drag it into position on the diagram. Set an element name and other properties as prompted.</p>	<p>Test Cut [1675]</p> <p>Test Set [1675]</p> <p>Test Suite [1676]</p>

Learn More:

- [Testpoint Management](#) [1672]

5.5.6 Diagram Tasks

This topic introduces many of the common tasks associated with managing diagrams.

Topic	Link
Add New Diagrams	Add New Diagrams [568]
Set Diagram Properties	Set Diagram Properties [569]
Add Elements and Connectors From Toolbox	Add Elements and Connectors From Toolbox [546]
Paste from the Project Browser	Paste from the Project Browser [576]
Copy And Paste Diagram Element	Copy And Paste Diagram Element [579]
Place Related Elements on Current Diagram	Place Related Elements on Current Diagram [580]
Delete a Diagram	Delete a Diagram [580]
Rename a Diagram	Rename a Diagram [581]
Change Diagram Type	Change Diagram Type [581]
Diagram Navigation Hotkeys	Diagram Navigation Hotkeys [582]
Copy Image to Disk	Copy Image to Disk [582]
Copy Image to Clipboard	Copy Image to Clipboard [583]
Duplicate a Diagram	Duplicate a Diagram [583]
Z Order Elements	Z Order Elements [584]
Set Default Diagram	Set Default Diagram [585]
Open a Package	Open a Package [585]
Feature Visibility	Feature Visibility [585]
Insert Diagram Properties Note	Insert Diagram Properties Note [587]

Topic	Link
Manage Legend Elements	Manage Legend Elements ^[588]
Autosize Elements	Autosize Elements ^[589]
Swimlanes	Swimlanes ^[590]
Swimlanes Matrix	Swimlanes Matrix ^[592]
Using the Image Manager	Using the Image Manager ^[593]
Show Realized Interfaces for a Class	Show Realized Interfaces for a Class ^[597]
Label Menu Section	Label Menu Section ^[597]
Pan and Zoom a Diagram	Pan and Zoom a Diagram ^[599]
Move Diagram Sections	Move Diagram Sections ^[600]
View Last and Next Diagram	View Last and Next Diagram ^[600]
Set Diagram Page Size	Set Diagram Page Size ^[600]
Scale Image to Page Size	Scale Image to Page Size ^[602]
Lock Diagram	Lock Diagram ^[602]
Lay Out a Diagram	Lay Out a Diagram ^[615]
Undo Last Action	Undo Last Action ^[603]
Redo Last Action	Redo Last Action ^[603]
Present Diagrams in a Model Views Slideshow	Present Diagrams in a Model Views Slideshow ^[477]

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Element permission to update or delete items on a diagram, and Manage Diagram permission to create, copy or delete diagrams

Learn More:

- [Permissions List](#) ^[205]

5.5.6.1 Add New Diagrams

This topic explains how to add a UML diagram, Extended diagram or MDG Technology diagram to a model in Enterprise Architect.

How to:

Step	Action	See Also
1	In the Project Browser, select the appropriate package or element under which to place the diagram	
2	Do one of the following: <ul style="list-style-type: none"> • In the Project Browser toolbar click on the New Diagram icon • Right-click to open the context menu and select the Add Add Diagram or Add Add <type> Diagram menu option • Press (Insert) and select the Add Add Diagram or Add Add <type> Diagram menu option, or • Select the Project New Diagram menu option 	

Step	Action	See Also
	The New Diagram dialog displays	
3	The Name field defaults to the name of the selected package or element; if necessary, type a different name for the new diagram	
4	In the Select From panel, click on the appropriate diagram category for the diagram The Diagram Types panel displays a list of the diagram types within the selected category	UML Diagrams [795]
5	In the Diagram Types panel, click on the type of diagram to create	
6	Click on the OK button to create your new diagram Enterprise Architect checks whether there is a template diagram for this diagram type in the Templates package and, if so, applies the display characteristics defined in that template	Set Element Templates Package [643]

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Manage Diagrams permission to create new diagrams
- The diagram type determines the default toolbar associated with the diagram and whether it can be set as a child of another element in the Project Browser (for example, a Sequence diagram under a Use Case)

Learn More:

- [Manage Diagrams](#) [205]

5.5.6.2 Diagram Properties

You can set several properties of a diagram using the diagram **Properties** dialog. Some properties influence the display and some are logical attributes that appear in the documentation.

Topics:

Topic	Detail	See Also
Usage	There are several options for opening the diagram Properties dialog for a given diagram: <ul style="list-style-type: none"> • Select the Diagram Properties menu option to open the Properties dialog for the currently active diagram • Right-click on the required diagram in the Project Browser and select the Properties context menu option 	

Topic	Detail	See Also
	<ul style="list-style-type: none"> Right-click on the background of the open diagram and select the Properties context menu option Double-click in the background of the open diagram. <p>In the Diagram Properties dialog you can set properties including name, author and version information, zoom factor, paper size and layout, diagram notes and various appearance attributes. Once you have made any necessary changes, click on the OK button to save and exit.</p>	

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Diagrams permission to change diagram properties
- You can also set the default diagram background color and the element fill color on the Standard Colors page of the Options dialog. You can set color gradients for both diagram background and element fill color on the Diagram Appearance page of the dialog

Learn More:

- [Update Diagrams](#) ^[205]
- [Standard Colors](#) ^[424]
- [Diagram Appearance](#) ^[427]
- [General Tab](#) ^[570]
- [Diagram Tab](#) ^[571]
- [Elements Tab](#) ^[573]
- [Features Tab](#) ^[574]
- [Connectors Tab](#) ^[576]

5.5.6.2.1 General Tab

The **General** tab of the diagram **Properties** dialog enables you to define characteristics of the overall diagram, such as its title, version and modification date.

Access: **Diagram | Properties > General**

Use to:

- Define characteristics of the overall diagram

Reference:

Field	Usage	See also
Name	Indicates the name of the diagram (defaults to the name of the parent package).	
Author	Specifies the name of the person who owns the diagram (defaults to the name of the person who created the diagram).	
Version	Indicates the version number of the diagram (defaults to 1.0).	

Field	Usage	See also
Stereotype	Indicates the name of the stereotype for the diagram. You can define stereotypes to select here using the Settings UML Types menu option, selecting the Stereotypes ^[1963] tab and creating stereotypes with a Base Class of Diagram .	Stereotypes Tab ^[1039]
Created	Automatically display the date the diagram was created.	
Modified	Indicates the date and time on which the diagram was last modified (defaults to the current date and time).	
Notes	Indicates any additional notes about the diagram. You can format the notes using the Notes toolbar at the top of the field.	Notes Toolbar ^[770]

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Diagrams permission to update diagram properties

Learn More:

- [Permissions List](#) ^[205]

5.5.6.2.2 Diagram Tab

The Diagram tab of the diagram Properties dialog enables you to define the representation of the diagram.

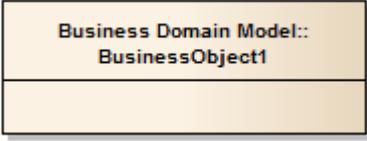
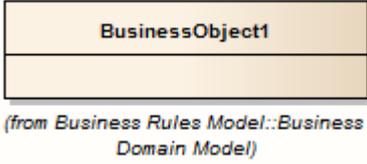
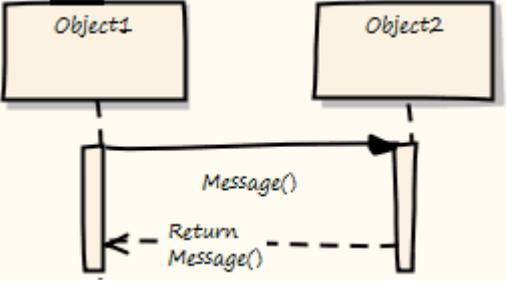
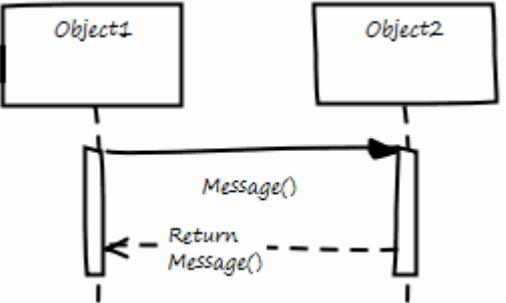
Access: **Diagram | Properties > Diagram**

Use to:

- Define the representation of a diagram

Reference:

Field	Usage	See also
Use Alias if Available	Display the element alias as the name if the alias is specified	
Show Additional Parents	Show the name of all parents not in the current diagram for all Classes and interfaces	
Show Page Border	Show a page border to align elements with	
Show Diagram Details	Show diagram details in a note in the top left corner of the diagram Deselect to hide the diagram details	
Show Sequence Notes	Show the Sequence Notes on the current diagram	
Show	Show the namespace of each element on the diagram, as part	

Field	Usage	See also
Namespace	of the element name; for example: 	
Fully Qualified Namespace	(If Show Namespace is selected) show the fully qualified namespace of each element on the diagram, under the element; for example: 	
Hand Drawn	Display the diagram contents as if they had been roughly drawn by hand; for example: 	
Whiteboard Mode	Displays the diagram as if it had been drawn on a whiteboard, with white fill and background; for example: 	
Disable fully scoped object names	Prevent the display of fully-qualified object names on the diagram	
Page Setup	Adjust the scaling of the image according to the size of the page it is to be printed on	Scale Image to Page Size ⁶⁰²
Print Page Header and Page Footer	Add page headers and footers to a print-out of the diagram; the headers and footers are generated from the diagram characteristics, such as the name of the creator and the date of modification	
Always Open as	Always display the diagram contents as a Diagram List rather	Diagram List ⁴⁶²

Field	Usage	See also
Element List	than as a diagram	
Always Open as Gantt	Always display the diagram contents as a GANTT chart	
Set Layout Style	Define the layout of the diagram, to be applied automatically when you select the Layout Diagram options	Layout Diagram ^[619]
RTF Document Options	Options for generating RTF reports for a particular diagram	Document Options ^[1729]
Exclude image from RTF documents	Exclude this diagram image from any RTF document generated on the parent package or element	
Document each contained element in RTF	Include documentation on each element in this diagram, in any RTF document generated on the parent package or element	Generate RTF Documentation ^[1729]
Divide Diagram into Multiple Pages	Divide each large diagram into separate pages in the RTF document	Scaled to page ^[602]
Rotate Images	Rotate each diagram image by 90 degrees in the RTF document	

Notes:

- The **Divide Diagrams into Multiple Pages** option is only effective when the **Scaled Printing** option is set to **None** on the Print Advanced dialog
- **Rotate Images** is only available for bitmap (.bmp) images

5.5.6.2.3 Elements Tab

The **Elements** tab of the diagram **Properties** dialog enables you to define what components of the elements should be displayed on the diagram.

Access: **Diagram | Properties > Elements**

Use to:

- Define what components of elements should be displayed

Reference:

Field	Usage	See also
Use Stereotype Icons	For elements that have whole shapes drawn by Enterprise Architect (such as Analysis stereotypes), draw the alternative shape (if defined). For elements that have an icon displayed in the top right corner, (such as an Artifact element) if Show Element Stereotypes is selected, display the stereotype icon instead of the stereotype text.	Analysis stereotypes ^[1186] Artifact ^[940]

Field	Usage	See also
Show Element Stereotypes	For elements that have whole shapes drawn by Enterprise Architect, if Use Stereotype Icons is deselected, display any stereotype on the element. For elements that have an icon displayed in the top right corner, indicate that a stereotype is present (icon if Use Stereotype Icons is selected, text if not).	
Show Table Owner	Display the Table Owner. For more information, see the Set Table Owner ^[1349] topic.	Set Table Owner ^[1349]
Show Element Property String	Show the advanced property string for all elements; for example, {leaf}.	
Show Compartments	Enable the following compartments to be shown or hidden for any element using rectangle notation: <ul style="list-style-type: none"> • Attributes • Operations • Tags (Tagged Values) • Requirements • Constraints • Testing (Testing Scripts^[1707]) • Maintenance (Maintenance Scripts)^[1715] • Package Contents • Notes 	Testing Scripts ^[1707] Maintenance Scripts ^[1717]

5.5.6.2.4 Features Tab

The Features tab of the diagram Properties dialog enables you to define how features (attributes and operations) are displayed on the diagram.

Access: **Diagram | Properties > Features**

Use to:

- Define how features (attributes and operations) are displayed on a diagram

Reference:

Field	Usage	See also
Show Qualifiers and Visibility Indicators	Show or hide the qualifiers and visibility indicators on the diagram <i>Qualifiers</i> include such things as the 'derived' symbol (<i>l</i>) and the public key symbol (PK) Visibility indicators ^[1457] include such things as +, -, # and ~, which indicate the scope of access of the item (such as an attribute, operation or role)	Visibility indicators ^[1457]
Show Stereotypes	Show the stereotypes on all features	
Show Property	Show the advanced property string for all element features, for	

Field	Usage	See also
String	example, {readOnly}	
Show Operation Return Type	Display the return data type of operations	
Suppress Brackets for Operations Without Parameters	Suppress brackets on operations that have no parameters; that is, Opn ; rather than Opn() ;	
Always Show Linked Features	Force display of linked attributes and operations, regardless of the setting of any other option that might hide them	Linked Attributes and Operations [734]
Visible Class Members	Hide Class members according to their scope and methods that specify properties	Visible Class Members [575]
Show Attribute Detail	Select whether to show both the attribute name and type, or the attribute name only	
Show Parameter Detail	Control the display of method parameters	Visible Class Members [575]

5.5.6.2.4.1 Visible Class Members

On the Features tab of the diagram Properties dialog, the **Visible Class Members** panel enables you to hide Class members by their scope and methods that specify properties. Use the check boxes to define the visibility of Class members.

Access: **Diagram | Properties > Features : Visible Class Members**

Use to:

- Define visibility of class members

Show Parameter Detail:

The **Show Parameter Detail** field enables you to control the display of method parameters with the following options:

Option	Effect
None	No details shown
Type Only	Shows the type of parameter only
Full Details	Shows all of the details for parameters
Name Only	Shows the name of the parameter only

5.5.6.2.5 Connectors Tab

The Connectors tab of the diagram Properties dialog enables you to define the appearance of the connectors on the diagram.

Access: **Diagram | Properties > Connectors**

Use to:

- Define the appearance of connectors on a diagram

Reference:

Field	Action	See also
Show Relationships	Show relationships in the current diagram	
Show Collaboration Numbers	Show numbering in Communication diagrams	
Show Non-Navigable Ends	Indicate when an Association end is not navigable; a cross is presented at the Association connector	
Show Connector Property String	Show the property string for connectors	
Suppress All Connector Labels	Hide all connector labels	
Show Stereotype Labels	Show (selected) or hide (deselected) any stereotype labels on the connectors on the diagram	
Connector Notation	Display the required connector notation: <ul style="list-style-type: none"> • UML 2.1 - use the standard UML 2.1 notation for connectors • Information Engineering - use the Information Engineering (IE) connection style • IDEX1 - use the Integrated Definition Methods IDEX1 connection style 	http://www.agiledata.org/essays/dataModeling101 http://www.idef.com/IDEF1X.html

5.5.6.3 Paste from Project Browser

As well as creating new elements in a diagram, you can drag existing elements from the **Project Browser** into the current diagram.

When you drag an existing element into the diagram, the **Paste Element** dialog displays to prompt you to select the type of paste action to carry out. (If the dialog does not display, press and hold (**Ctrl**) while dragging the element onto the diagram.)

Topics:

Topic	Detail	See also
Pasting Options	<p>Three paste options are available:</p> <p>Paste the element as a simple link. In this case the original element exists both in the current diagram and in the original source diagram. Changes to the element are reflected in all diagrams in which it is shown.</p> <p>Paste as an instance of the element. If the element can have instances such as an Object, Sequence instance or Node instance, you can drop the element in as an instance of the source element, with the classifier pre-set to the original source. This is useful when creating multiple instances of a Class in a Sequence diagram or Communication diagram.</p> <p>If you select this option, the Copy connectors checkbox is enabled. If you select this checkbox, any connectors between the original element and any other elements that have also been pasted to this diagram are reproduced as connectors between the instances.</p> <p>Create as a child of the source element. This automatically creates a new Class - which you are prompted to name - with a Generalization connector back to the source. This is very useful when you have a Class library or framework from which you inherit new forms; for example, you can paste a Hashtable as "MyHashtable" which automatically becomes a child of the original Hashtable. Used with the Override parent operations feature, this is a quick way to create new structures based on frameworks such as the Java SDK and the .NET SDK.</p> <p>You can make your selection on this dialog the default for:</p> <ul style="list-style-type: none"> • all drag and drop operations, or • only those where you display this Paste Element dialog <p>If you select the This Dialog checkbox, you should then select the Only show this dialog when (Ctrl) +Mouse drag is used checkbox and, on the Diagram Behavior page of the Options dialog, the Auto Instance checkbox.</p> <p>The effect of these selections is to give you two default paste options:</p> <ul style="list-style-type: none"> • Just drag the element onto the diagram and automatically create an instance • Press (Ctrl) while you drag the element from the Project Browser, displaying the Paste Element dialog, and click on the OK button to automatically paste the element according to whatever option you last selected from the dialog <p>If you select the All Drag and Drop checkbox on the Paste Element dialog, this deselects the Auto Instance checkbox on the Options dialog and enables you to add existing elements to the diagram according to the paste option you selected, without pressing (Ctrl) and without displaying a dialog. (If you want to change the default paste option, press (Ctrl) as you drag to display the dialog again and make your changes.)</p>	Object Classifiers Override Parent Operations Diagram Behavior

Notes:

- When an element dragged from the Project Browser is pasted into the diagram, its default style and size is applied rather than any alternative styles and sizes used for instances of the element in other diagrams

Learn More:

- [Connect Requirements](#)^[116]
- [Create Object From Attribute](#)^[70]
- [Make Linked Element A Local Copy](#)^[64]

5.5.6.3.1 Paste Multiple Items

You can paste multiple elements from the **Project Browser** into the current diagram.

Topics:

Topic	Detail	See also
Pasting Multiple Elements	<p>To select multiple elements, click on the selected items from the Project Browser while pressing and holding:</p> <ul style="list-style-type: none"> • (Ctrl) to add single items to the selection of multiple elements, or • (Shift) to select all the elements between the first and last selected items in the Project Browser. <p>You can then drag the selected elements from the Project Browser onto the current diagram, pressing and holding (Ctrl) ; for each element you have selected, the Paste Element dialog displays, prompting you to select the type of paste action to carry out.</p>	Pasting Options ^[57]

5.5.6.3.2 Paste Composite Elements

When you drag a Composite element from the **Project Browser** onto the current diagram, Enterprise Architect prompts you to select the type of paste action to carry out with the Composite element.

(If the dialog does not display, press and hold (**Ctrl**) while dragging the element onto the diagram.)

Topics:

Topic	Detail	See also
Pasting composite elements	<p>Two advanced options are available for pasting Composite elements; these require the include Embedded Elements checkbox to be selected:</p> <ol style="list-style-type: none"> 1. The All Embedded Elements option, which pastes all of the Composite element's embedded elements. 2. The Based on instance option, which pastes only the elements contained in a specific instance of the Composite element, maintaining the layout of the embedded elements. <p>Click on the drop-down arrow and select the appropriate instance.</p> <p>For details of the other options on this dialog, see the Paste from Project Browser topic.</p>	Paste from Project Browser ^[57]

5.5.6.3 Paste Activities

You can paste an Activity from the **Project Browser** into the current diagram.

Topics:

Topic	Detail	See also
Pasting Activities	<p>When you drag an Activity from the Project Browser onto the current diagram, The Paste Element dialog displays, prompting you to select the type of paste action to carry out. (If the dialog does not display, press and hold (Ctrl) while dragging the element onto the diagram.)</p> <p>Two options are available:</p> <ul style="list-style-type: none"> • Paste the Activity as a link: in this case the Activity appears in the current diagram as a simple reference to the original source Activity. Changes to the Activity in the diagram affect all other links to this Activity • Paste as an invocation of the Activity; if you select this option, the Copy connectors checkbox is enabled. If you select this checkbox, any connectors between the original Activity and any other elements that have also been pasted to this diagram are reproduced as connectors between the instances <p>For details of the other options on this dialog, see the Paste from Project Browser topic.</p>	<p>Paste from Project Browser 576</p>

Learn More:

- [Activity](#) 573

5.5.6.4 Copy And Paste Diagram Element

How to:

To copy a diagram element, follow the steps below:

Step	Instruction	See Also
1	Select the element(s) to copy.	
2	For multiple elements, right-click to open the context menu and select the Copy menu option. Alternatively, press (Ctrl+C) .	
3	For single elements, select the Edit Copy menu option or alternatively press (Ctrl+C) .	

To paste diagram elements, follow the steps below:

Step	Instruction	See Also
1	Open the diagram to paste into.	
2	Right-click on the diagram background to open the diagram context menu.	

Step	Instruction	See Also
3	Select either the Paste Object(s) as New menu option (completely new element) or the Paste Object(s) as Link menu option (reference to the existing element).	

Notes:

- The Date Created and Time Created parameters of a pasted-as-new element are set to the current date and time; the parameters for a linked element remain the same as the copied element

5.5.6.5 Place Related Elements on Diagram

To find and place related elements on the current diagram, use the Relationships window.

Access: [View | Relationships](#)

Topics:

Topic	Detail	See Also
Usage	<p>Right-click on any connector in the list to open the context menu.</p> <p>If an element is not present in the current diagram, the context menu contains the Place Target Element in Diagram option. This is useful when you are building up a picture of what an element interacts with, especially when reverse engineering an existing code base.</p> <p>Select the Place Target Element in Diagram option. Move the cursor to the required position in the diagram and click to place the element. Alternatively, press (Esc) to cancel the action.</p>	

5.5.6.6 Delete Diagram**How to:**

To delete a diagram from your model, follow the steps below

Step	Action	See Also
1	In the Project Browser, right-click on the diagram to delete The context menu displays	
2	Select the Delete '<diagram name>' menu option. A confirmation prompt displays.	
3	Click on the OK button to confirm the delete.	

Notes:

- In Enterprise Architect there is no Undo feature for deleting diagrams, so be certain that you want to delete a diagram before you do so
- When you delete a diagram, you do not delete the elements in the diagram from the model
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to create new

diagrams

- You can also delete multiple diagrams from the Project Browser, by holding (**Ctrl**) or (**Shift**) while you select them, then right-clicking on one of them and selecting the Delete selected items context menu option

Learn More:

- [Permissions List](#)^[205]

5.5.6.7 Rename Diagram

To rename a diagram, follow the steps below

How to:

Step	Action	See Also
1	Open the Diagram Properties dialog by double-clicking on the diagram background, or by selecting the Diagram Properties menu option.	
2	In the Name field on the General tab, type the new name for your diagram.	
3	Click on the OK button to save changes.	

5.5.6.8 Change Diagram Type

If necessary, you can change one type of diagram to another type. This is useful if you have either made a mistake in selecting the diagram type to begin with, or if the purpose and nature of a diagram changes during analysis.

Access: **Diagram | Advanced | Change Type**

How to:

To change a diagram type, follow the steps below:

Step	Action	See Also
1	Open the diagram to change.	
2	Select the Change Type menu option The Change Diagram Type dialog displays	
3	Select the required diagram type	
4	Click on the OK button to save changes	

Notes:

- Some diagram types do not transfer to others; for example you cannot change a Class diagram into a Sequence diagram

5.5.6.9 Diagram Navigation Hotkeys

The diagram hotkeys enable you to quickly navigate to and select elements within a diagram. The following table details the key combinations and their functionality.

Hotkey Command	Action
Shift + Arrow , Element(s) selected	Move the selected element(s) by increments.
Arrow , No element selected	Scroll around the diagram.
Esc	Clear the current selection.
Tab	Select the first element in the diagram if none currently selected.
Shift + click	Add the clicked element to the current selection.
Ctrl + click	Add the clicked element to the current selection.
Ctrl + Shift + drag	Pan the diagram.
Alt + G	Select the item in the Project Browser and give it focus.

5.5.6.10 Copy Image to Disk

You can copy a diagram image to a disk file in the following formats:

- Windows bitmap (256 color bitmap)
- GIF image
- Windows Enhanced Metafile (standard metafile)
- Windows Placeable Metafile (older style metafile)
- PNG format
- JPG
- TGA

How to:

To copy a diagram image to file, follow the steps below:

Step	Action	See Also
1	Open the diagram to save.	
2	Select the Diagram Save as Image menu option, or press (Ctrl+T) .	
3	When prompted, enter a name for the file and select an image format.	
4	Click on the OK button.	

Notes:

- Enterprise Architect clips the image size to the smallest bounding rectangle that encompasses all diagram elements

5.5.6.11 Copy Image to Clipboard

You can copy diagram images onto the MS Windows clipboard and paste them directly into MS Word or other applications.

To copy an image to the clipboard, follow the steps below

How to:

Step	Action	See Also
1	Open the diagram to copy.	
2	Select the Diagram Copy Image to Clipboard menu option, or press (Ctrl+B)	
3	Click on the OK button. The diagram has been copied to the clipboard and can now be pasted into compatible applications or into another diagram. You can set the clipboard format on the Options dialog (Tools Options menu option, General page). Enterprise Architect supports bitmap or metafile format.	General Settings ⁴²²

5.5.6.12 Copy (Duplicate) Diagram

Enterprise Architect makes it easy to duplicate a complete diagram, with either :

- links back to the original diagram elements (*shallow mode*) or
- complete copies of all elements in the diagram (*deep mode*) or
- complete copies of all elements in the diagram that have the same parent as that of the diagram and links back to the original diagram elements for all other elements on the diagram (*smart mode*)

To duplicate a diagram, follow the steps below

How to:

Step	Action	See Also
1	In the Project Browser , select the diagram to copy.	
2	Right-click to display the context menu and select the Copy Diagram menu option.	
3	Navigate to the package to host the new diagram, and right-click to open the context menu.	
4	Select the Paste Diagram menu option. The Copy Diagram dialog displays.	
5	In the Name field, type the name for the new diagram.	
6	In the Type of copy panel, click on the radio button for the type of copy you require; either linked elements (<i>shallow mode</i>) or complete copies of the original elements shown on the diagram (<i>deep mode</i>) or complete copies of the original elements shown on the diagram that have the same parent as that of the diagram and linked elements of all other elements on the diagram (<i>smart mode</i>) .	
7	Click on the OK button. Enterprise Architect automatically creates the new diagram, links or creates new	

Step	Action	See Also
	elements and arranges them as in the original diagram. All links are also copied between diagram elements where appropriate.	

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Manage Diagrams permission to copy diagrams.
- You can also paste a copied diagram as a child of a composite element.
- For a diagram copied in *shallow mode*, any changes to the properties of the original elements are reflected in the elements in the pasted diagram.
- For a diagram copied in *deep mode*, any changes to the properties of the original elements are not reflected in the elements in the pasted diagram.
- For a diagram copied in *smart mode*, any changes to the properties of the original elements are reflected only in those elements that are copied as simple links in the pasted diagram.
- Element position and size will be independent in all copy modes.

Learn More:

- [Permissions List](#)^[205]
- [Composite Elements](#)^[647]

5.5.6.13 Z Order Elements

Z Order refers to an element's depth in the diagram perspective, and thus influences which elements appear in front of others and which appear behind.

How to:

To set the Z Order of an element, follow the steps below:

Step	Action	See Also
1	Right-click on the element in the diagram	
2	Select the Z-Order menu option	
3	The displayed commands enable you to move the selected element(s) back, forward, to the front of all other elements or behind all other elements	

Notes:

- This option enables you to change the Z-order of a single element approximately in relation to the rest of the diagram; you can also select the **Modify Z-Order** option from the *diagram* context menu, to review and adjust the Z-Order of each element on the diagram precisely relative to each other element

Learn More:

- [Diagram Context Menu](#)^[538]

5.5.6.14 Set the Default Diagram

A project might have a default diagram. If set, this diagram loads when Enterprise Architect first opens the model. It is often convenient to place hyperlinks to other diagrams and resources on the default diagram, thus creating a Home Page for your model.

Access: [Diagram](#) | [Advanced](#) | [Make Model Default](#)

Notes:

- Once you have specified a default diagram, the **Home** icon on the **Diagram** toolbar takes you back to that diagram from your current location in the model

5.5.6.15 Open Package From Diagram

How to:

To open a package from within a diagram follow the steps below:

Step	Action	See Also
1	Open a diagram that shows the package to open.	
2	Right-click on the package element to open the context menu.	
3	Select the Open Package option. Alternatively, press (Ctrl+K) .	

Notes:

- Enterprise Architect finds the package default diagram and opens it for you. This is the first available diagram in the package, selected in alphabetical order; for example, a diagram called Alpha in a child package or element several levels down opens before a diagram called Beta immediately under the selected package

5.5.6.16 Feature Visibility

Enterprise Architect enables you to set the visibility of attributes and operations - where shown - for selected elements on a specific diagram only. You can hide or show attributes and operations by scope, or you can hide attributes and operations individually. The visibility you set applies only to the current diagram, so a Class can appear in one diagram with all features displayed, and in another with features hidden. For example, you can hide all protected attributes, all private operations or any other combination of attributes and operations.

It is possible to show inherited attributes, operations, requirements, constraints and Tagged Values for elements that support those features. When Enterprise Architect displays inherited features, it creates a merged list from all generalized parents and from all realized interfaces. If a child Class redefines something found in a parent, the parent feature is omitted from the Merge List.

How to:

To customize feature visibility, follow the steps below:

Step	Action	See Also
1	Either: <ul style="list-style-type: none"> • Click on the element in the diagram and either click on the Element Feature Visibility menu option or press (Ctrl+Shift+Y), or 	

Step	Action	See Also
	<ul style="list-style-type: none"> Right-click on the element in the diagram to display the context menu and click on the Feature Visibility option. <p>The Feature Visibility dialog displays.</p>	
2	To filter display of attributes or operations by scope, select the checkbox against each scope that should be visible and clear the checkbox against each scope that should not.	
3	<p>In the Show Element Compartments panel, select the compartments to display for the element on the diagram.</p> <p>The Fully Qualified Tags checkbox enables you to display the full provenance of a Tagged Value, where the same Tagged Value can be used several times in different contexts with different values. The description in the Tagged Value compartment reads:</p> <p><Profile>::<Stereotype>::<Tagged Value name>=<Value> for example: BPMN::Activity::Activity Type = Task.</p> <p>(Only for Tagged Values created in Enterprise Architect release 7.1 or later.)</p> <p>If you select the Notes checkbox, the Notes compartment on the element in the diagram displays the text that has been typed into the Notes field of the Element Properties dialog. This checkbox also enables the maximum chars field, which defaults to 1000 as the number of characters of notes text displayed. Overtyping this value to display less text or more text.</p> <p>The change only applies to the selected elements on the diagram, so you can display full notes for a selected element whilst the other elements on the diagram have no notes text.</p>	
4	<p>In the When Resizing Elements panel, select the appropriate option for resizing the Class, object or table to prevent very wide diagram objects.</p> <p>The selected option defaults to Resize to longest Feature, so that the minimum width for a diagram object is determined by its longest displayed attribute, method or other compartment value. If necessary, you can change the option to Wrap Features (so that any longer features are wrapped onto multiple lines) or Truncate Features (so that longer features are not displayed in full).</p>	
5	If required, in the Inherited Features panel, select one or both checkboxes to set whether Enterprise Architect should display inherited features as well as directly owned ones.	
6	Click on the OK button to save changes. Enterprise Architect redraws the diagram with the appropriate level of feature visibility.	

Topics:

Topic	Detail	See also
Suppress or Show Specific Features	<p>The Custom button in the Attribute Visibility and Operation Visibility panels enables you to show or hide specific operations and attributes. If you select the Show checkbox, the Custom button displays the Show Features in Diagram dialog; if you deselect the checkbox, the button displays the Suppress Features in Diagram dialog.</p> <p>The two dialogs are identical, but in the first you select the checkboxes of specific features to show, and in the second you select the</p>	

Topic	Detail	See also
	checkboxes of specific features to hide. You can also use the Filter by Scope button in this dialog to, for example, list only operations that are Protected and select, say, two of them to hide, so that on the diagram the element displays all but two of the Protected operations and all operations of other scopes.	

Notes:

- To show features for element types that do not have visible compartments, such as Use Cases and Actors, right-click on the diagram object to display the context menu and select the **Advanced Settings | Use Rectangle Notation** option
- The **Show** checkbox, if selected, overrides these selections to display all attributes or operations in the element, except those specifically deselected in the Show Features in Diagram dialog
- If you have selected the **Notes** checkbox, you can select the **Render Formatted Notes** checkbox to display the text on the diagram, formatted using the Notes toolbar

Learn More:

- [Notes](#)^[77]

5.5.6.17 Insert Diagram Properties Note

Properties of a diagram can be displayed on screen within a custom text box. You can move this text box around and change its appearance. You cannot change what the text box says.

Topics:

Topic	Detail	See also
Usage	To create the note, drag the Diagram Notes element from the Common page of the Toolbox onto the diagram. Alternatively, select the Diagram Property Note menu option, or click on the Diagram Properties Note button on the UML Elements toolbar and click on the diagram 	

Notes:

- This is not the same as the diagram details note, which displays in the top left corner of the diagram if the Show Diagram Details checkbox is selected on the Diagram Properties dialog. You cannot move the diagram details, nor change the appearance. To hide the diagram details, deselect the checkbox

Learn More:

- [Configure Default Appearance](#)^[64]
- [Diagram Properties](#)^[57]

5.5.6.18 Create Legends

A *Legend* shape identifies colors and styles you have used to group other elements on the diagram. You can use the Legend to assist in distinguishing different elements, connectors or systems on the diagram. For example, the Legend could show that all elements concerned with the management system are shaded in blue, and all outcomes connectors are shown in red. The Legend displays as a key to the diagram, with the filled shape styles first and the lines and connector styles underneath.



You add a Legend to the diagram, then edit it to add Legend elements, which define the colors and styles used in the diagram.

To add a Legend to a diagram, drag the *Diagram Legend* element from the **Common** page of the **Toolbox** onto the diagram (or click on the **New Diagram Legend** icon () on the **UML Elements** toolbar, and click on the diagram).

To edit the Legend either double-click on the Legend or Right-click on the Legend and select the **Properties** context menu option.

5.5.6.18.1 Legend Properties

Access: drag the Diagram Legend element from the Common page of the Toolbox onto the diagram

Use to:

- Add a Legend to a diagram

Reference:

Field	Usage	See also
Name	Indicates a name for an element displayed within a legend.	
Fill Color	Indicates a fill color for an element within a legend.	
Line Color	Indicates a line color for an element within a legend.	
Line Thickness	Denotes the line thickness of an element within a legend.	
Style Options	Allows further alteration options to the legend appearance.	Style Options ⁵⁸⁹
Fill	Indicates that a fill element is to be created/modified.	
Line	Indicates that a line element is to be created/modified.	

Hand (Up)	If more than one element is present within a legend, this button moves a highlighted element up one entry within the description list.	
Hand (Down)	If more than one element is present within a legend, this button moves the highlighted element down one entry within the description list.	
New	Add a new element to the legend.	
Save	Saves an element and adds it to the description list.	
Delete	Deletes an element from the description list.	
Description List	Specifies the current elements created within the legend.	

Notes:

- The Legend dialog enables you to add, delete, modify or re-sequence Legend elements. Use the **Fill** tab to define the Legend elements for shapes, then click on the **Line** tab to define Legend elements for lines and connectors

5.5.6.18.2 Style Options

Click on the **Style Options** button (...) to display the Style Options dialog, on which you can modify a Legend title, font size, background color and border color. If you choose default options for the colors, the Legend automatically assumes colors based on the diagram background color.

Access: Legend dialog > (...)

Use to:

- Modify appearance options for a Legend

Reference:

Field	Usage	See also
Legend Title	Indicates the title of the Legend	
Heading Size	Indicates the size of the font for a Legend	
Font Color	Indicates the font color the Legend is to use	
Background	Indicates the background color the Legend is to use	
Border Color	Indicates the border color the Legend is to use	

5.5.6.19 Autosize Elements

You can autosize an element or group of elements in a diagram to the default size for the element type (for a Class, 90 x 70 pixels at 100% zoom). However, if the element contains more information than the default size can show (such as a long name, long attributes or additional compartments) the autosize option resizes the element to the minimum size for revealing the information.

The size change effectively operates around the mid point of each element, so the layout and size of the diagram do not change. Automatically changing the layout of a diagram is a different process.

How to:

To autosize elements, follow the steps below:

Step	Action	See Also
1	Select the elements to resize (press (Ctrl+A) to select all).	
2	Either: <ul style="list-style-type: none"> • Right-click on any of the elements and, on the context menu, select the Autosize menu option, or • Press (Alt+Z) . 	

Notes:

- Not all elements resize: elements such as Events remain the same; Timing and Sequence diagrams (where position is crucial) are unchanged; and elements added from a profile or Shape Script maintain any size definitions imposed by the profile
- With an element image created with a Shape Script that contains a defSize command, **Autosize** returns the element to the defSize value and not the element default size

Learn More:

- [Lay Out a Diagram Automatically](#)^[619]
- [defSize command](#)^[1094]

5.5.6.20 Swimlanes

Enterprise Architect diagrams support *Swimlanes* for all diagram types. Swimlanes are vertical or horizontal bands in a diagram that divide the diagram into logical areas or partitions. In the example below the activities relating to particular entities within the model (such as the User, or the back end Repository) are placed within a containing swim lane to indicate their association.

Access: **Diagram | Swimlanes and Matrix > Swimlanes**

Use to:

- Add Swimlanes to a diagram

Reference:

Field	Usage	See also
Active	Indicates that the Swimlane dialog is active, as opposed to the Matrix dialog	Swimlanes Matrix ^[592]
Orientation	Specifies the orientation of the swimlane, either vertical or horizontal	
Line Color	Indicates the line color for the swimlane	
Font Color	Indicates the font color for the swimlane	

Line Width	Indicates the line width of the swimlane	
Locked	Indicates that the swimlane can not be moved on the diagram	
Bold Font	Indicates that the name given to the swimlane is set to bold	
Hide Classifier	Indicates that the classifier name is not shown on the diagram	
Hide Names	Indicates that the names set for swimlanes do not appear on the dialog	
Show Names in Title Bar	Specifies that the name given to a swimlane appear in the title bar	
Hand (Up)	If more than one element is present within a swimlane, this button moves a highlighted element up one entry within the Swimlanes list	
Hand (Down)	If more than one element is present within a swimlane, this button moves a highlighted element down one entry within the Swimlanes list	
New	Opens the Swimlane Details dialog, allowing specification of the Name, Classifier and Background Color	Swimlane Details ^[59]
Modify	Opens the Swimlane Details dialog, allowing modification of the Name, Classifier and Background Color	Swimlane Details ^[59]
Delete	Deletes a Swimlane from the Swimlanes List	
Swimlanes List	The list of the currently defined Swimlanes	

Notes:

- If you set a background color for a swimlane, it takes on the same shading profile as the main diagram background

5.5.6.20.1 Swimlane Details

The Swimlane Details dialog opens if the **New** or **Modify** buttons are clicked from the Swimlanes dialog.

Access: **Diagram | Swimlanes and Matrix > Swimlanes : New/Modify**

Use to:

- Specify details about a swimlane

Reference:

Field	Usage	See also
Name	Indicates a name for the swimlane.	
Classifier	Indicates the classifier name for the swimlane, the (...) button allows specification of a classifier.	Select <Item> Dialog ^[69]
Back Color	Specifies a background color for a swimlane.	

5.5.6.21 Swimlanes Matrix

Enterprise Architect diagrams support a *Swimlanes Matrix* for all diagram types, based on the Zachman Framework.

The Swimlanes Matrix divides the diagram into cells of vertical columns and horizontal rows. The cell in the top left corner of the Swimlanes Matrix contains the heading of the matrix. The first cell at the top of each column contains the column title text. The first cell at the left of each row contains the row title text.

Access: [Diagram | Swimlanes and Matrix > Matrix](#)

Use to:

- Add a Matrix to a diagram

Reference:

Field	Usage	See also
Active	Indicates that the Matrix dialog is active, as opposed to the Swimlanes dialog.	Swimlanes ^[590]
Title #	Indicates a title name, up to 3 title names can be specified.	
Color	Indicates a font color for a title.	
Font	Opens the font dialog, font styles can be specified here.	Set Element Font ^[657]
Hidden	Indicates if a title is hidden.	
Type	Indicates what type of matrix heading is created.	
Back Color	Indicates what background color the matrix item has.	
Hide Item	Indicates that the matrix item is hidden.	
Model Profiles	Indicates what profile is used during creation of the Matrix.	
Model Profiles : Save	Saves a profile to the Model Profiles list Prompts for a profile name.	
Model Profiles : Delete	Deletes a profile from the Model Profiles list.	
Lock	Indicates that the matrix item can not be moved or resized on a diagram.	
Line Widths	Indicates the width on lines surrounding a matrix.	
Hand (Up)	If more than one item is present within a matrix, this button moves a highlighted element up one entry within the Matrix List.	
Hand (Down)	If more than one element is present within a matrix, this button moves a highlighted element down one entry within the Matrix List.	

New	Allows for the creation of a new matrix item.	
Copy	Copies the highlighted matrix item.	
Delete	Deletes a matrix item from the Matrix List.	
Matrix List	A list of currently defined matrix items.	

Notes:

- The heading is the first item in the list; you create only one heading
- When you define columns and rows, you define the header or title cells. The properties of these cells do not reflect on the matrix cells themselves. For example, the intersection cell of a column and row has a transparent background and therefore takes the color and shading effect of the diagram background
- You can also transport all the matrix profiles between models (as Diagram Matrix Profiles), using the Export Reference Data and Import Reference Data options
- By applying a Model Profile, you replace the current profile. Save the current profile to avoid losing it
- To size the rows and columns, drag the row and column borders on the diagram. Elements placed inside each cell are shifted when sizing. To prevent the elements shifting, press and hold (**Ctrl**) while sizing

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.5.6.22 Using the Image Manager

The **Image Manager** dialog enables you to insert alternative images in diagrams, rather than inserting standard UML elements. For example, you might want to place a **custom background** image on a diagram, or display a custom image such as a Router or PC on a UML element.

Access: **Right-click on an element within a diagram | Appearance | Select Alternate Image (Shortcut - Ctrl + Shift + W)**

To locate and display an image, click on individual image filenames, or press (**↑**) and (**↓**) to scroll through the list of images. As you highlight each image filename, the Preview panel changes to reflect the image. Double-click on the required image filename to display the image in full size.

On the Image Manager dialog, the following buttons are available:

Reference:

Action	Usage	Shortcut	See also
View	Display the selected image in full size.	Alt + V	
Usage	Display the Element Usage ^[654] dialog, which lists the diagrams in which the selected image is used.		Element Usage ^[630]
Rename	Change the filename of the selected image. The option first warns you that the change would impact the other elements that use this image, and prompts you to confirm that you want to go ahead with the name change.		

Action	Usage	Shortcut	See also
	If you continue, a prompt displays for you to enter the new filename.		
Add New	Browse appropriate directories to search for and import new images. You can import images in .BMP, .PNG, .EMF, .WMF, .TGA, .PCX or .JPG format. Internally, Enterprise Architect stores the images in .PNG or metafile format to conserve space.	Alt + A	
Update Selected	Refresh the selected image; for example, after it has been modified.	Alt + U	
Delete	(Not available for an image from a deployed MDG technology.) Delete the selected image. A message displays to indicate how many elements use the image. Click on the Continue button to delete information about the image from those elements, which then revert to their previous appearance.	Alt + D	
Cancel	Close the Image Manager dialog.		
OK	Confirm selection of the alternative image for the element selected in the diagram.	Alt + O	

Notes:

- For elements with lifelines, such as those used on Sequence diagrams, the Lifeline must remain intact to enable messages to be created between the elements. Therefore such elements cannot have alternative images.
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Configure Images permission to configure alternative element images
- There are two categories of image available through the Image Manager: those images that have been imported into the model, which are stored in the model; and those images that are incorporated into a deployed MDG technology, which are not copied into the current model but are held in memory. The technology images are identified by the technology name followed by the image file name; for example: MDGFrame::ovalshape.emf
- If you are creating many elements of the same type that have a particular image, you should use a custom stereotype with an associated metafile
- You can transport image files between models, using the Export Reference Data and Import Reference Data options

Learn More:

- [Custom background image](#) ^[595]
- [Permission List](#) ^[205]
- [Adding Images in MDG Technology](#) ^[107]
- [Custom Stereotype](#) ^[1018]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

5.5.6.22.1 Create Custom Diagram Background

Enterprise Architect diagrams have a single-color 'wash' background that you can set to a solid color or a fade gradient down the screen. You set the color on the Standard Colors page of the Options dialog, and whether to have a fade gradient on the Diagram Appearance page.

Alternatively, using the Image Manager dialog, you can create a non-tiled background for diagrams.

How to:

To create a non-tiled background follow the steps below:

Step	Action	See Also
1	Create a Boundary object from the Use Case Elements page of the Toolbox; do not use the Boundary element from any other section of the Toolbox	System Boundary ^[93] Toolbox ^[546]
2	Stretch the Boundary to a size that can contain all of the elements you intend to place on the diagram, and drag it to the edges of the diagram workspace	
3	Right-click on the Boundary element The context menu displays	
4	Select the Z-Order Send to Bottom menu option; this ensures that the Boundary is not displayed in front of any other element in the diagram	
5	Either: <ul style="list-style-type: none"> • Press (Ctrl+Shift+W), or • Right-click on the Boundary to display the context menu, and select the Appearance Alternate Image menu option 	
6	On the Image Manager dialog, select an appropriate image as the diagram background and ensure that the image size is large enough to span the required size of the diagram background	Image Manager ^[593]
7	When you have selected the required image, click on the OK button Alternatively, you can copy an image from another source onto the Windows clipboard, right-click on the Boundary element in the Enterprise Architect diagram, and select the Appearance Apply Image From Clipboard context menu option	

Learn More:

- [Standard Colors](#)^[424]
- [Diagram Appearance](#)^[427]

5.5.6.22.2 Import Image Library

Using the Image Library enables you to create attractive diagrams with custom images. A bundled clip art collection of UML-based images is available as an Imported Image Library, from the Sparx Systems website. Image libraries enable you to import a collection of images into the Image Manager in one process.

How to:

To import the Image Library, follow the steps below (before starting, you must have a suitable Image Library file):

Step	Action	See Also
1	Download the Image Library from the Sparx Systems website.	www.sparxsystems.com/resources/image_library.html
2	Select the Project Model Import/Export Import Reference Data menu option. The Import Reference Data dialog displays.	
3	Locate the XML Image Library file to import using the Select File button. The file name is ImageLibrary.xml in the directory in which you saved the file.	
4	Select the data set containing the Image Library. Then click on the Import button.	

To use the images contained within the Image Library, follow the steps below

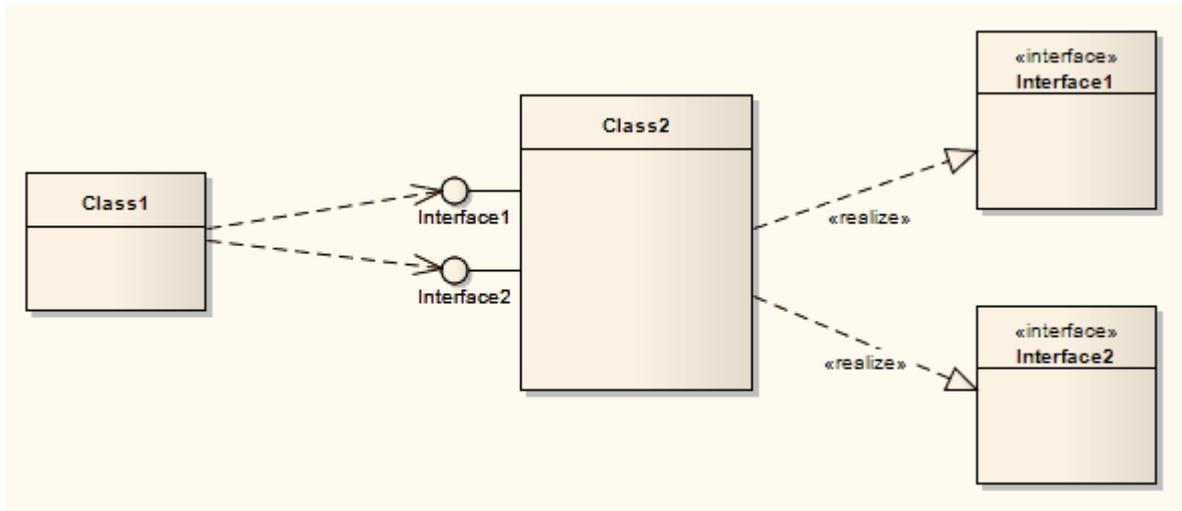
Step	Action	See Also
1	Create a diagram to associate with the images contained in the Image Library.	
2	Select the element to change from the default appearance to one of the images contained within the library.	
3	Press (Ctrl+Shift+W) , or right-click on the selected element to display its context menu and then select the Appearance Select Alternate Image option.	
4	On the Image Manager dialog, in the Name field highlight the appropriate image name and then click on the OK button.	

Notes:

- Images contained within the Image Library are copyright of Sparx Systems, are only available for use in conjunction with Enterprise Architect, and are supplied on the understanding that they are not used under any other circumstance

5.5.6.23 Show Realized Interfaces of Class

You can display each interface directly realized by a Class as a 'lollipop' style interface node, which protrudes from the left-hand side of the Class. Connectors can be directly attached to the node, indicating usage of the interface part of the Class or component. See the example below:



Topics:

Topic	Detail	See also
Explanation	<p>In this example, Class2 realizes Interface1 and Interface2 as represented by the interface nodes protruding from the Class. Class1 is dependent on these two interfaces, which is shown by the Dependency connectors linking to the nodes.</p> <p>To show nodes for the interfaces a Class realizes, as in the above diagram, right-click on the Class and select the Embedded Elements Show Realized Interfaces context menu option. This setting only applies to the selected Class, and can be changed at any time.</p>	

5.5.6.24 Label Menu Section

You can add labels to both connectors and elements, using the element or connector context menu as follows:

- Element:
 - Select the Embedded Elements menu option and either the **Add <element>** option or the **Embedded Elements** option; the label is the embedded element name
 - Apply an alternative image to an element (that might also have a run state; the run state, attributes and operations of the element are then displayed as a label of the element
 - Connector - Select the Properties option and define the connector name, stereotype, constraints and/or source and target roles

Once you have these labels, you can edit and format them using the **Labels** context menu.

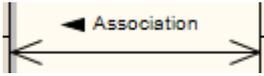
Access: Right-click on a label

Reference:

The **Labels** context menu associated with embedded elements provides the following options:

Action	Usage	Shortcut	See also
Set Label Color	Specify a color for the label.		
Hide Label	Hide the label; to unhide the label, right-click on the element and select the Appearance Show labels [655] context menu option.		Appearance Menu section [655]
Bold	Set the label font to bold.		
Text Alignment	Align the text within the label text area. The options available from the submenu enable you to specify left, center and right alignment.		
Label Rotation	Orient the label in the horizontal or vertical planes, with the vertical plane offering the option of clockwise or anti-clockwise position.		
Default Position	Move the label to the initial default location.		
Default Color	Set the label color to the default color.		

The **Labels** menu associated with connectors provides the following options:

Action	Usage	Shortcut	See also
Set Label Color	Specify a color for the label.		
Hide Label	Hide the label; to unhide the label use the Visibility Set Label Visibility [752] option on the connector context menu.		Set Label Visibility [752]
Bold	Set the label font to bold.		
Text Alignment	Align the text within the label text area. The options available from the submenu enable you to specify left, center and right alignment.		
Label Rotation	Orientate the label horizontally or vertically and, if vertically, in a clockwise or anti-clockwise position.		
Direction	<p>Set a small arrow at the end of the label pointing to either the label source or the destination dependent upon selection from the available options.</p>  <p>This is part of the label, so if there is no label there is no direction indicator.</p>		
Default Position	Move the label to the default location.		
Default Color	Set the label color to the initial default color.		

Notes:

- As labels can be concentrated on and around the element or connector, make sure that you click on a section of the required label that is clear of any other label or structure

Learn More:

- [Embedded Elements](#) ^[653]
- [Using the Image Manager](#) ^[593]
- [Run-time State](#) ^[953]
- [Connector Properties](#) ^[756]

5.5.6.25 Pan and Zoom a Diagram

Topics:

Topic	Detail	See also
Pan	<p>Pan the Diagram View in the following ways:</p> <ul style="list-style-type: none"> • Use (←), (→), (↑), (↓), (Page Up), (Page Down), (Home) and (End) when the Diagram View is selected • Use the scrollbars • Use the middle mouse button • Use the Pan & Zoom window 	<p>Pan & Zoom Window ^[473]</p>
Zoom	<p>You can zoom into and out from a diagram using the zoom buttons on the diagram toolbar, or by using the Diagram Zoom submenu.</p>  <p>Change the zoom level by 10% by clicking on either the Zoom In (+) or Zoom Out (-) buttons. Alternatively, select the Zoom In or Zoom Out options from the Diagram Zoom submenu.</p> <p>There are three ways to return the diagram to 100%:</p> <ul style="list-style-type: none"> • Click on the  button • Select Zoom to 100% from the Diagram Zoom submenu • (Ctrl) +middle-click the mouse 	

Notes:

- You can zoom in and out of the main window dynamically by holding (**Ctrl**) and rolling the mouse wheel
- Changes in diagram magnification through the zoom options can be saved as permanent changes to the diagram
- At high levels of zoom, element features cease to display. This is because of the difficulty the Windows font mapper has in choosing a font for extreme conditions, and the result can look odd

5.5.6.26 Move Elements In Diagram Sections

As you build up a diagram, you might find that you have to move part of the diagram up, down or to one side. You can do this in one of two ways:

Topics:

Topic	Detail	See also
Moving Elements	<ul style="list-style-type: none"> Hold the left mouse button down and drag over a group of elements to move (creating an outline around the elements), then click on an element in the outline and move the group as required Press (Alt) and click on the diagram, then drag the cursor to move everything beyond the cursor in the direction of the movement. <p>The first method enables you to reposition groups of elements within the larger diagram. The second method enables you to create space within the diagram without pushing some elements into others, as might happen if you cannot see the whole diagram on one screen.</p> <p>When you press (Alt) and click on the diagram, as you move the cursor a line displays on the diagram just behind the cursor. If you are moving the cursor left, everything to the left of the line moves with the cursor. If you move the cursor up, everything above the line moves up.</p> <p>However, if you move the cursor diagonally, two lines display to create a quadrant, and everything within the quadrant moves. For example, if you move the cursor left and down, everything below and left of the cursor moves.</p>	
Fine Movement	To adjust (or 'nudge') the position of a single element or a selected group of elements, press (Shift) + (→), (←), (↑) or (↓).	

5.5.6.27 View Last and Next Diagram

Enterprise Architect enables you to step backwards and forwards through the currently-open diagrams, including the **Start Page**.

Topics:

Topic	Detail	See also
Usage	 <p>To view the previous or next diagram use the Previous or Next buttons on the Diagram toolbar.</p> <p>Use the Home button to display the default project diagram (if one has been specified).</p>	Default Project Diagram ^[585]

5.5.6.28 Set Diagram Page Size

You can change the size of the diagram area (or scrollable/printable area) using the Diagram Properties dialog.

How to:

To set the page size, follow the steps below:

Step	Action	See Also
1	Load a diagram.	
2	Double-click on the background to open the Diagram Properties dialog.	
3	Click on the Diagram tab and, in the Appearance panel ensure that the Show Page Border checkbox is selected.	
4	On the Page Setup panel, click on the Advanced button. The Print Advanced dialog displays.	
5	Click on the Page Setup button. The Page Setup dialog displays. As you adjust the settings on this dialog, the page icon at the top illustrates the effects of your changes.	
6	In the Size field, click on the drop-down arrow and select an appropriate page size.	
7	In the Orientation panel click on the radio button for the orientation of the page to print.	
8	In the Margins panel, type the required left, right, top and bottom page margins for the diagram, in inches.	
9	Click on the OK button on the Page Setup dialog, the Print Advanced dialog, and the Diagram Properties dialog. The area within the page boundary lines on your diagram is expanded or reduced accordingly. When you print or print preview, the output is cropped to these boundary lines and the diagram divided between the necessary number of pages.	

Topics:

Topic	Detail	See also
Setting the Default Paper Size for New Diagrams	You can set the default paper size for new diagrams on the Diagram page of the Options dialog (select the Tools Options Diagram menu option). Once the paper size is set there, all new diagrams have that as the default size.	Configure Local Options - Diagram ^[426]

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Diagrams permission to change diagram page setup

Learn More:

- [Permissions List](#)^[205]

5.5.6.29 Scale Image to Page Size

When you print a diagram, the default setting is to scale the image to fit the size of the printer paper you have defined in the page set-up. The image is not scaled up to fill the page, but it is scaled down if it exceeds the current page boundary. The image retains its current proportions; that is, it is scaled down equally in the X and Y dimensions. For a large diagram, this can mean that the components of the diagram are small and hard to read.

Alternatively, you can print a multi-page image; that is:

- allow the diagram image to print on as many printer pages as it naturally occupies, (no scaling), or
- scale the diagram image to exactly fit a specified number of pages

In all three cases you also define the paper size and orientation.

How to:

To turn off or customize image scaling options, follow the steps below:

Step	Action	See Also
1	Select the diagram to scale	
2	Either: <ul style="list-style-type: none"> • Double-click on the diagram background, or • Right-click on the background and select the Properties context menu option The <type> Diagram: <name> dialog displays	
3	Click on the Diagram tab and, in the Page Setup panel click on the Advanced button The Print Advanced dialog displays	Page size and Alignment ^[60]
4	Select from the following options as required: <ul style="list-style-type: none"> • None: select to print on as many pages as the diagram image covers • Scale to 1 page: select to scale the diagram image to fit on the currently selected page • Custom: select to specify the width and height of the diagram images across a specified number of pages • Page Setup: click to select the page size and alignment 	

Notes:

- Before printing, make sure you have selected the required page layout using the **Page Setup** button

Learn More:

- [File Menu](#)^[74]

5.5.6.30 Lock Diagram

You can lock a diagram against inadvertent changes, such as moving or sizing elements.

How to:

To lock a diagram, follow the steps below:

Step	Action	See Also
1	Open the diagram to lock	
2	Right-click on the background to open the diagram context menu	
3	Click on the Lock Diagram option to prevent further changes	
4	Click on the OK button If a user selects an item on a locked diagram, the object border or outline displays in red	

Notes:

- This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions if security is enabled. In that case, see the Lock Model Elements topic

Learn More:

- [Lock Model Elements](#)^[210]

5.5.6.31 Undo Last Action

When editing diagrams, Enterprise Architect supports multiple undo levels for moving, re-sizing and deleting elements, and for deleting connectors.

Topics:

Topic	Detail	See also
Usage	There are three ways to undo the last action: <ul style="list-style-type: none"> • Press (Ctrl+Z) • Select the Edit Undo menu option • Click on the Undo button in the Default Tools toolbar -  	

Notes:

- Currently you cannot undo element additions or connector moves

5.5.6.32 Redo Last Action

When editing diagrams, Enterprise Architect supports multiple undo levels for moving, re-sizing and deleting elements, and for deleting connectors. If an Undo action is in error, you can redo the action to reverse the Undo.

Topics:

Topic	Detail	See also
Usage	There are three ways to redo the last action: <ul style="list-style-type: none"> • Press (Ctrl+Y) 	

Topic	Detail	See also
	<ul style="list-style-type: none"> Select the Edit Redo menu option Click on the Redo button in the Default Tools toolbar -  	

5.5.7 Layout Diagrams

Enterprise Architect provides a Layout Tools window to enable you to lay out the elements in a diagram. Unless the diagram is very simple, this facility does not lay out the entire diagram; it consists of a set of tools to set out different areas or sets of elements in the diagram.

How to:

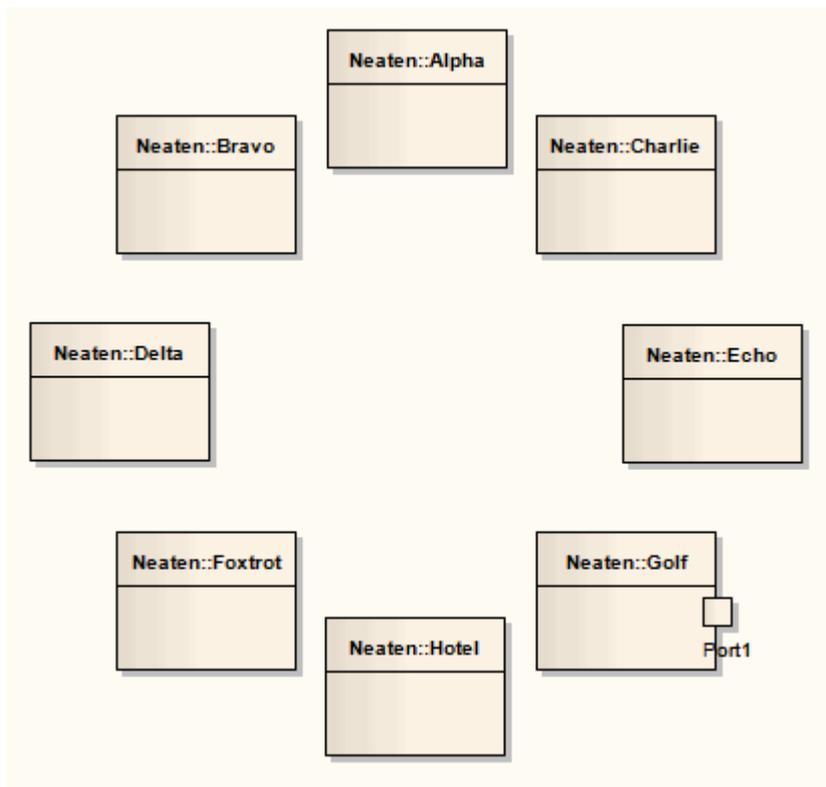
To start to lay out a diagram, follow the steps below:

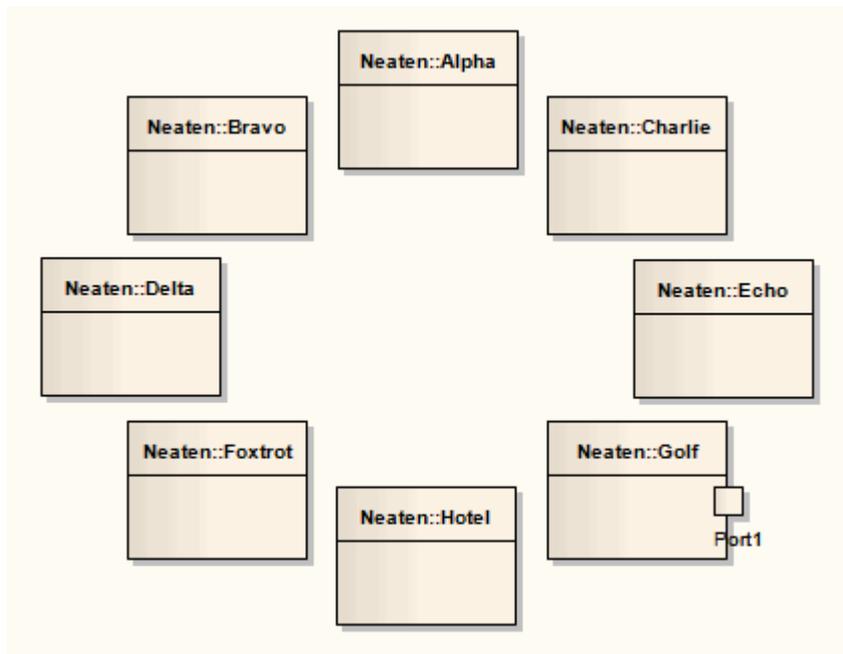
Step	Action	See Also
1	Open the Layout Tools window; either: <ul style="list-style-type: none"> Select the View Diagram Layout menu option, or Right-click on the main toolbar and select the Layout Tools context menu option 	
2	Select the elements to lay out on the currently-active diagram - hold (Shift) or (Control) while you click on each required element, or hold the mouse button down while you sweep over the area containing the required elements If no elements are selected on the active diagram, then all elements on the diagram are laid out (except where otherwise documented)	
3	Click on the drop-down arrow on the top left field of the Layout Tools window, and select the required layout type	
4	The layout type determines the fields presented in the window, therefore the appropriate fields are described in the topic for each layout type <ul style="list-style-type: none"> Circle/Ellipse Box Per Page Digraph Spring Neaten Converge/Diverge Fan Relations Auto Route 	Circle/Ellipse ^[605] Box ^[609] Per Page ^[611] Digraph ^[612] Spring ^[613] Neaten ^[614] Converge/Diverge ^[615] Fan Relations ^[617] Auto Route ^[618]
5	When you have completed the fields, click on the  button Enterprise Architect sets out the selected elements according to the options you have selected	
6	If you do not want to work with the new layout, click on the Undo button in the toolbar ()	Automatically Laying out a Diagram ^[619]

Step	Action	See Also
	Enterprise Architect also provides a facility for automatically laying out a diagram; if necessary, you can manually adjust the final result of this automatic process	

5.5.7.1 Circular/Elliptical Layout

The *Circle* and *Ellipse* layouts arrange the selected elements in a circle or elliptical pattern, using the largest horizontal and vertical element edge in the set of elements when calculating the radius of the layout arc.



**How to:**

To invoke the Circular or Ellipse layout, follow the steps below:

Step	Action	See Also
1	Follow the general Layout Diagrams procedure, and at step 3 select either Circle or Ellipse as required	Layout Diagrams 604
2	Click on the drop-down arrow in the Sort By: field and select the required sort parameter: <ul style="list-style-type: none"> • None - Elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom) • Area (Ascending) - Elements are passed to the specified layout in order of the screen space they occupy, smallest to largest • Area (Descending) - Elements are passed to the specified layout in order of the screen space they occupy, largest to smallest • Name (Ascending) - Elements are passed to the specified layout in alphanumeric order, based on the element name • Name (Descending) - Elements are passed to the specified layout in reverse alphanumeric order, based on the element name • Element Type - Elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name 	
3	Under the Placement option, select either: <ul style="list-style-type: none"> • Top to Bottom (the elements are positioned in the required order, zig-zagged across the perimeter of the circle or ellipse - see Diagram A, below) • Circular (the elements are placed in the required order, clockwise around the perimeter of the circle or ellipse - see Diagram B, below) 	
4	Select the Center focused element checkbox to put the last-selected element (the one with the hashed border) in the center of the circle or ellipse - see Diagram C, below	

Diagram A - Top To Bottom Layout

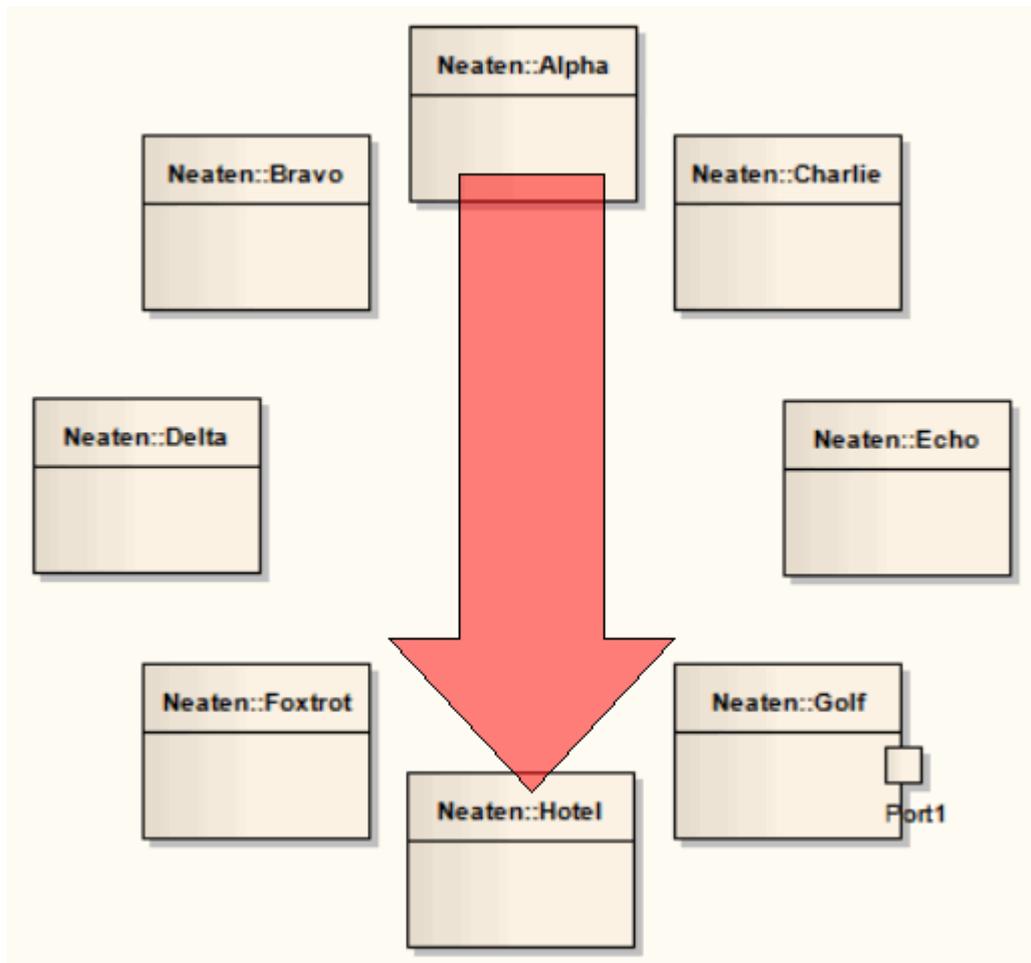


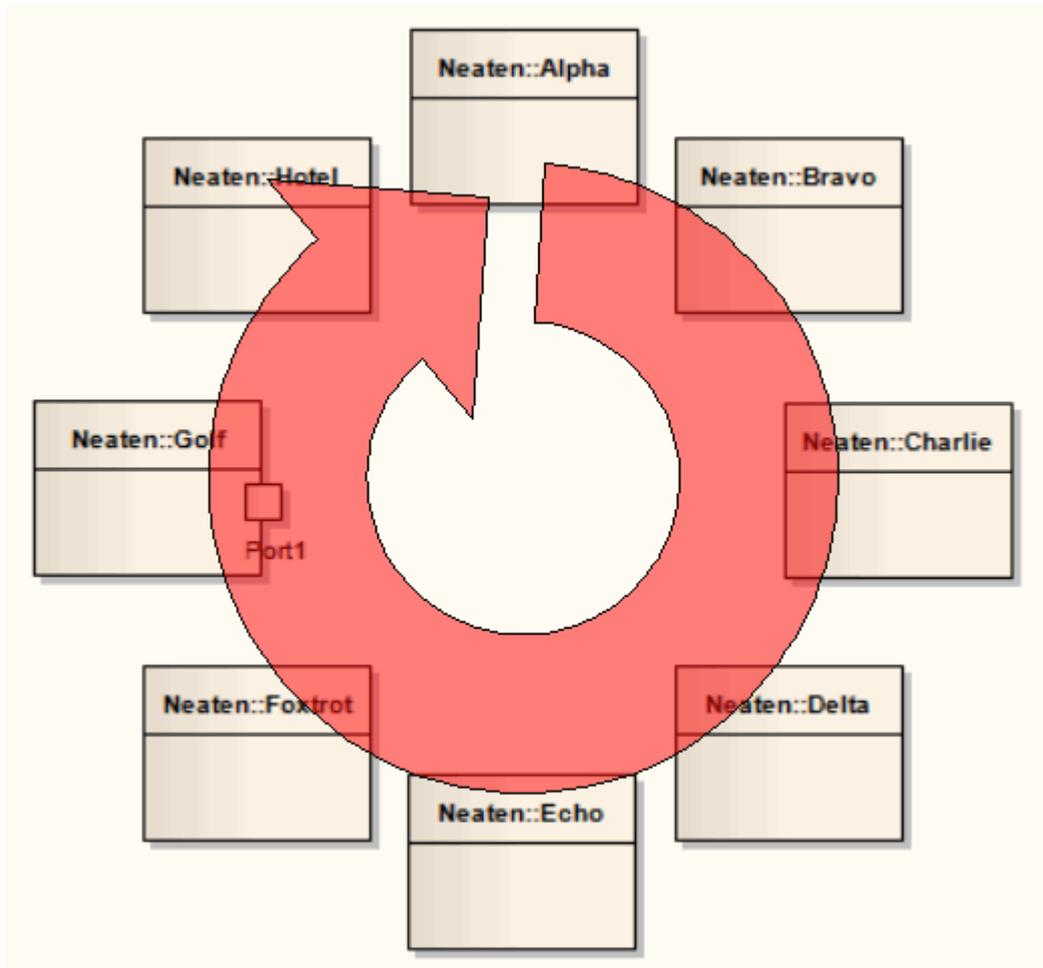
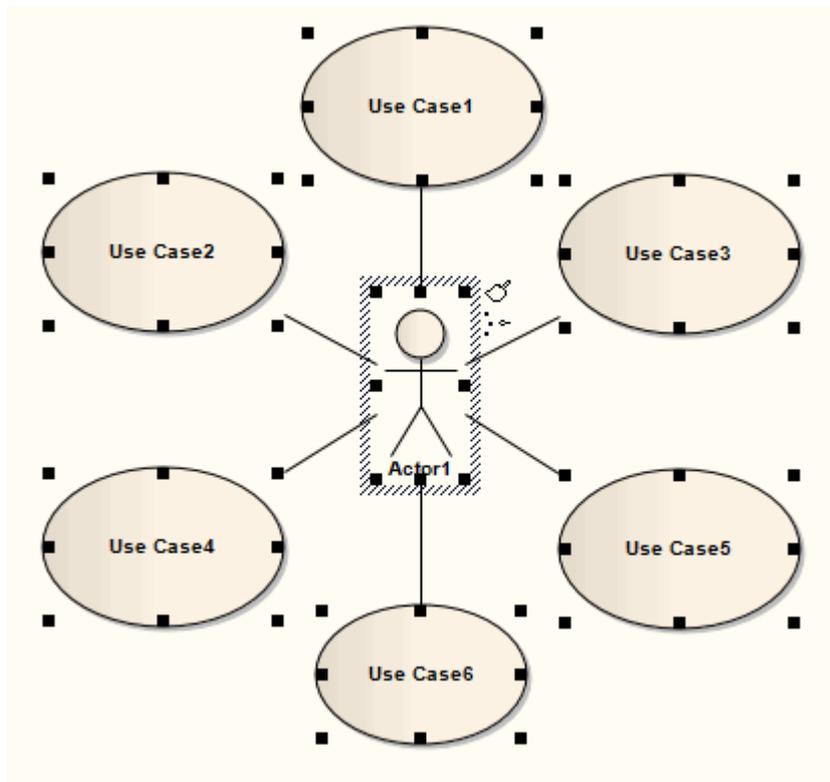
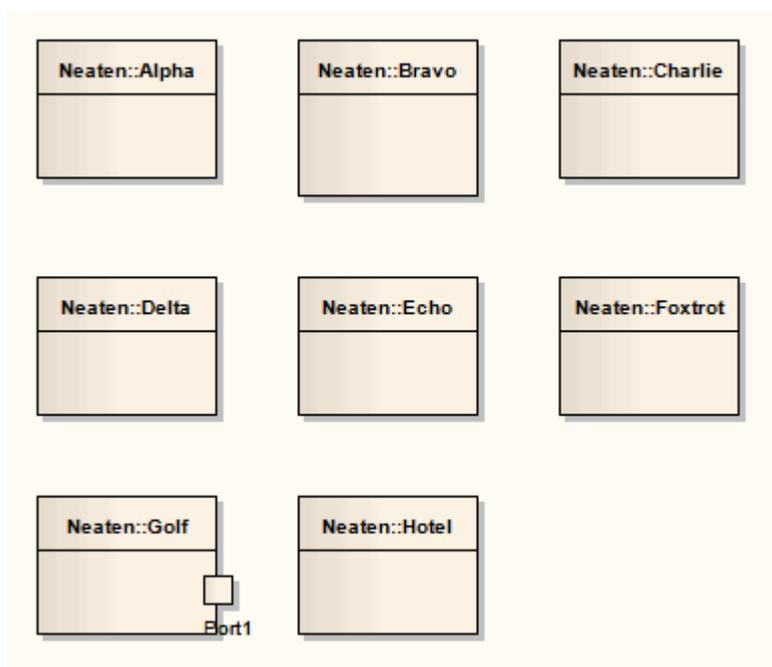
Diagram B - Circular Layout

Diagram C - Center Focused Element



5.5.7.2 Box Layout

The *Box* layout arranges the set of selected elements into a square grid.



The dimensions of the box are determined by the square root of the number of selected elements (for example, 16 elements create a 4x4 box).

How to:

To invoke the Box layout, follow the steps below:

Step	Action	See Also
1	Follow the general Layout Diagrams procedure, and at step 3 select Box	Layout Diagrams 604
2	Click on the drop-down arrow in the Sort By: field and select the required sort parameter: <ul style="list-style-type: none"> • None - Elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom) • Area (Ascending) - Elements are passed to the specified layout in order of the screen space they occupy, smallest to largest • Area (Descending) - Elements are passed to the specified layout in order of the screen space they occupy, largest to smallest • Name (Ascending) - Elements are passed to the specified layout in alphanumeric order, based on the element name • Name (Descending) - Elements are passed to the specified layout in reverse alphanumeric order, based on the element name • Element Type - Elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name 	
3	In the Padding (px) field, type the vertical and horizontal distance between elements, in pixels	
4	Select the appropriate element distribution option: <ul style="list-style-type: none"> • Automatically distribute: Automatically calculate the dimensions of the box (the square root of the number of selected elements; for example, 16 elements create a 4x4 box) • Specify distribution: Manually define the width of the box, in columns 	
5	If you selected Specify Distribution , in the Columns field type the required number of columns	

5.5.7.3 Per Page Layout

The *Per Page* layout divides each diagram page into a number of cells, which house the selected elements. The number of cells per page is determined by the page distribution parameter, as explained below.



How to:

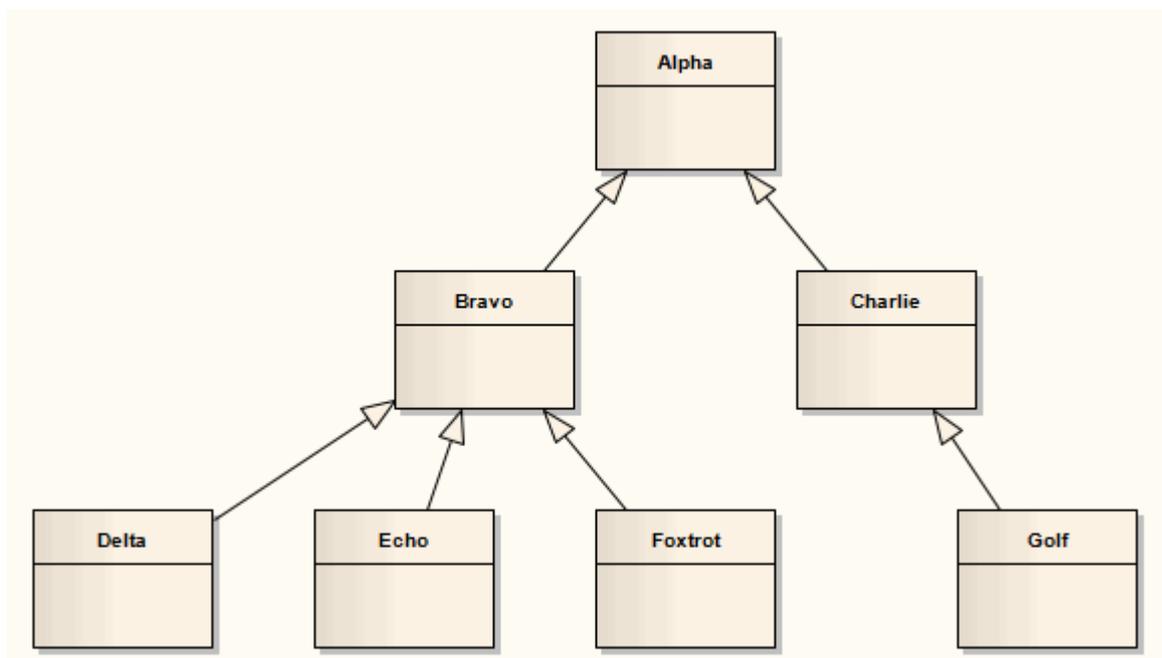
To invoke the *Per Page* layout, follow the steps below:

Step	Action	See Also
1	Follow the general Layout Diagrams procedure, and at step 3 select Per Page	Layout Diagrams ⁶⁰⁴
2	Click on the drop-down arrow in the Sort By: field and select the required sort parameter: <ul style="list-style-type: none"> None - Elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom) Area (Ascending) - Elements are passed to the specified layout in order of the screen space they occupy, smallest to largest Area (Descending) - Elements are passed to the specified layout in order of the screen space they occupy, largest to smallest Name (Ascending) - Elements are passed to the specified layout in alphanumeric order, based on the element name Name (Descending) - Elements are passed to the specified layout in reverse alphanumeric order, based on the element name Element Type - Elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name 	
3	In the Padding (px) field, type the vertical and horizontal distance between cells, in pixels	

Step	Action	See Also
4	Select the appropriate page distribution option: <ul style="list-style-type: none"> • Automatically distribute: Automatically calculate the optimum number of cells, taking into consideration the largest horizontal and vertical element edges • Specify distribution: Manually enter the per page grid dimensions 	
5	If you selected Specify Distribution , in the Rows and Columns fields type the required number of rows and columns	
6	Select the Center Elements checkbox to place each element in the center of its cell; otherwise the element placement defaults to the top left corner of the cell	
7	In the Start Page field, type the number from which to start page numbering; pages begin at the top left and continue horizontally to the right	

5.5.7.4 Digraph Layout

The *Digraph* layout arranges the selected elements into a directed graph (digraph for short). The Digraph attempts to highlight the hierarchy of the elements while keeping the direction of all connectors pointing to the same edge of the diagram.



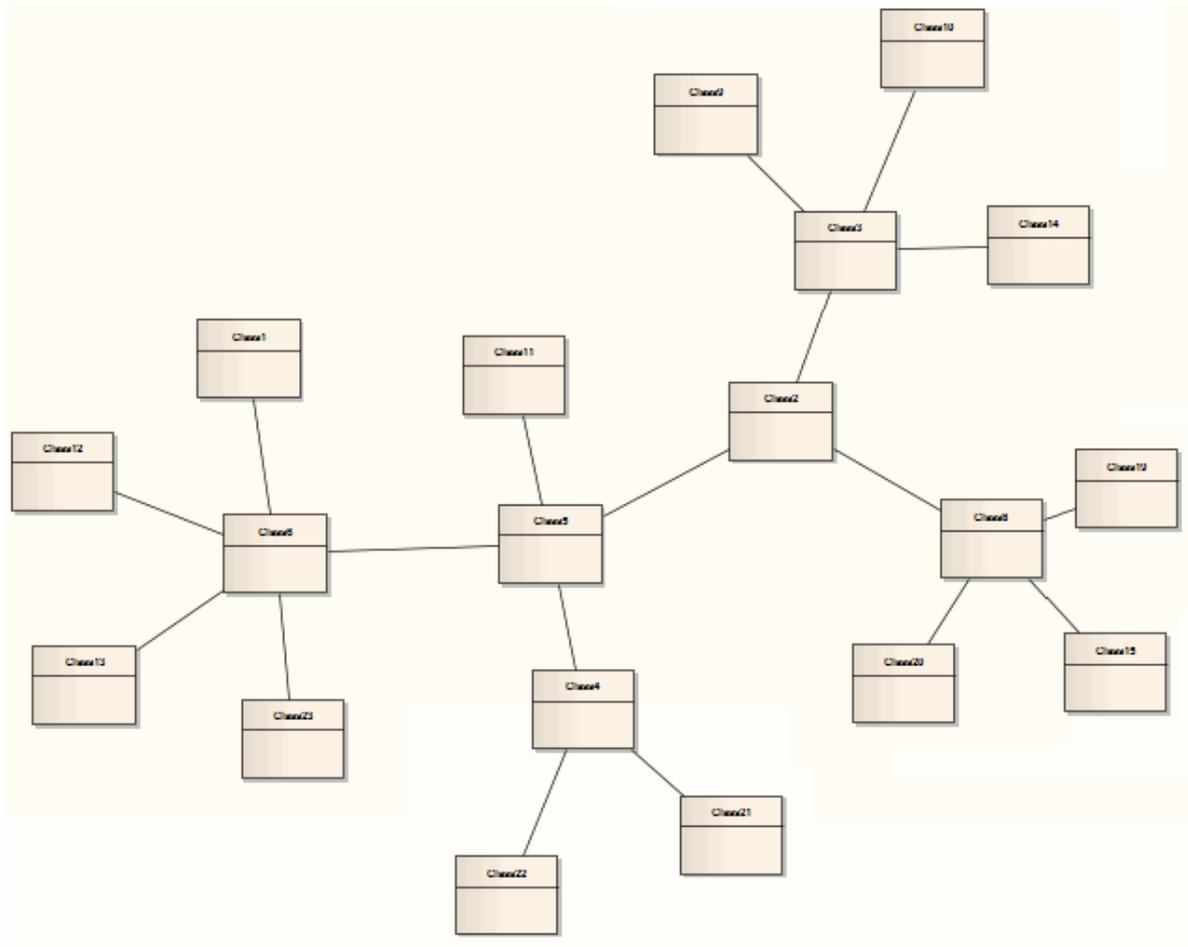
The Digraph layout provides the same behaviour as the Automatic Diagram layout.

Learn More:

- [Layout Diagrams](#)^[604]
- [Lay out a Diagram Automatically](#)^[619]

5.5.7.5 Spring Layout

The *Spring* layout uses a force-directed approach to arrange the selected elements organically.



The Spring layout employs a physical analogy to lay out elements. Each element is treated as a particle with a like electrical charge that repels other elements. Connectors act as springs (hence the term Spring layout) that draw connected elements back together. The layout is good for highlighting clusters of related objects and identifying symmetry in the graph.

How to:

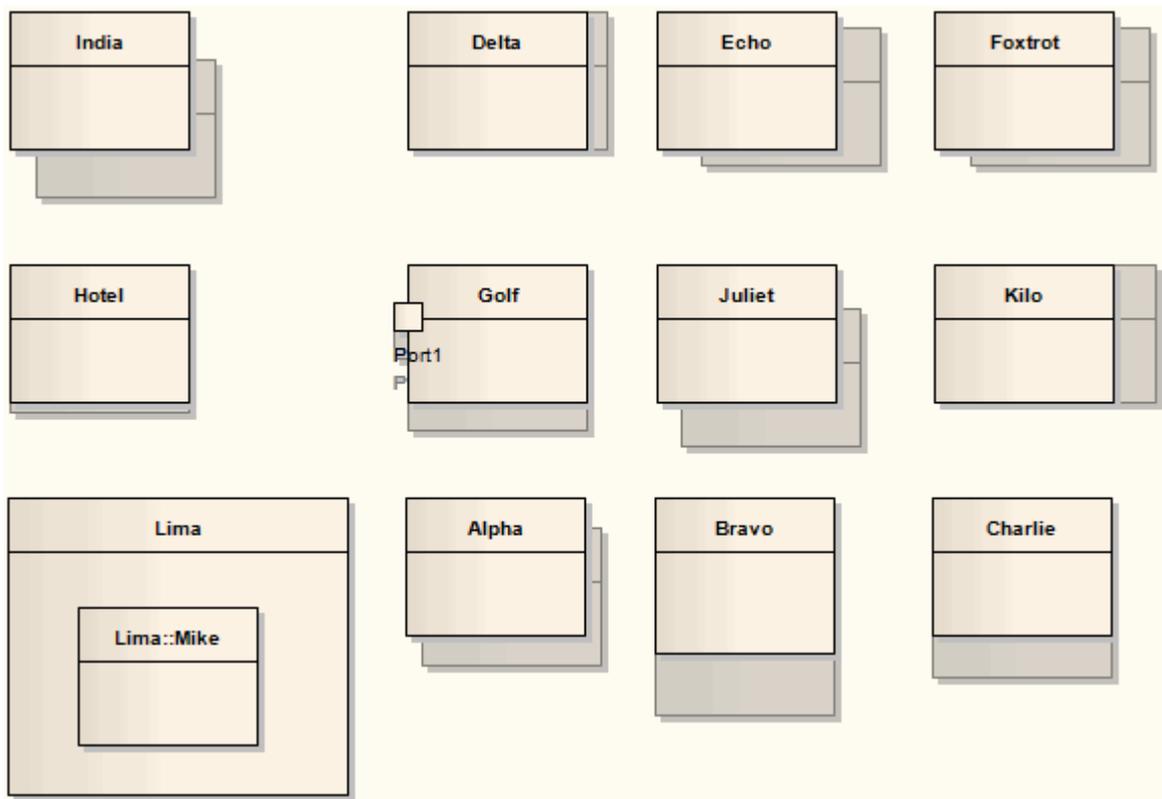
To invoke the Spring layout, follow the steps below:

Step	Action	See Also
1	Follow the general Layout Diagrams procedure, and at step 3 select Spring	Layout Diagrams ⁶⁰⁴
2	In the Iterations field, type the number of iterations, or rounds, to perform to reach the final layout The layout is developed over several iterations; depending on the complexity of the graph, increasing the number of iterations produces a better result but takes longer to calculate	

Step	Action	See Also
3	If the diagram contains elements that significantly vary in size, and that might overlap in the final layout, select the Scale to prevent overlap checkbox to scale up the positions of the selected elements (preserving size) until no elements overlap	

5.5.7.6 Neaten Layout

The *Neaten* layout attempts to arrange the selected elements into a grid based on their horizontal and vertical proximity to each other. Elements that share the same row or column are aligned based on the **Column** and **Row Snap** parameters.



How to:

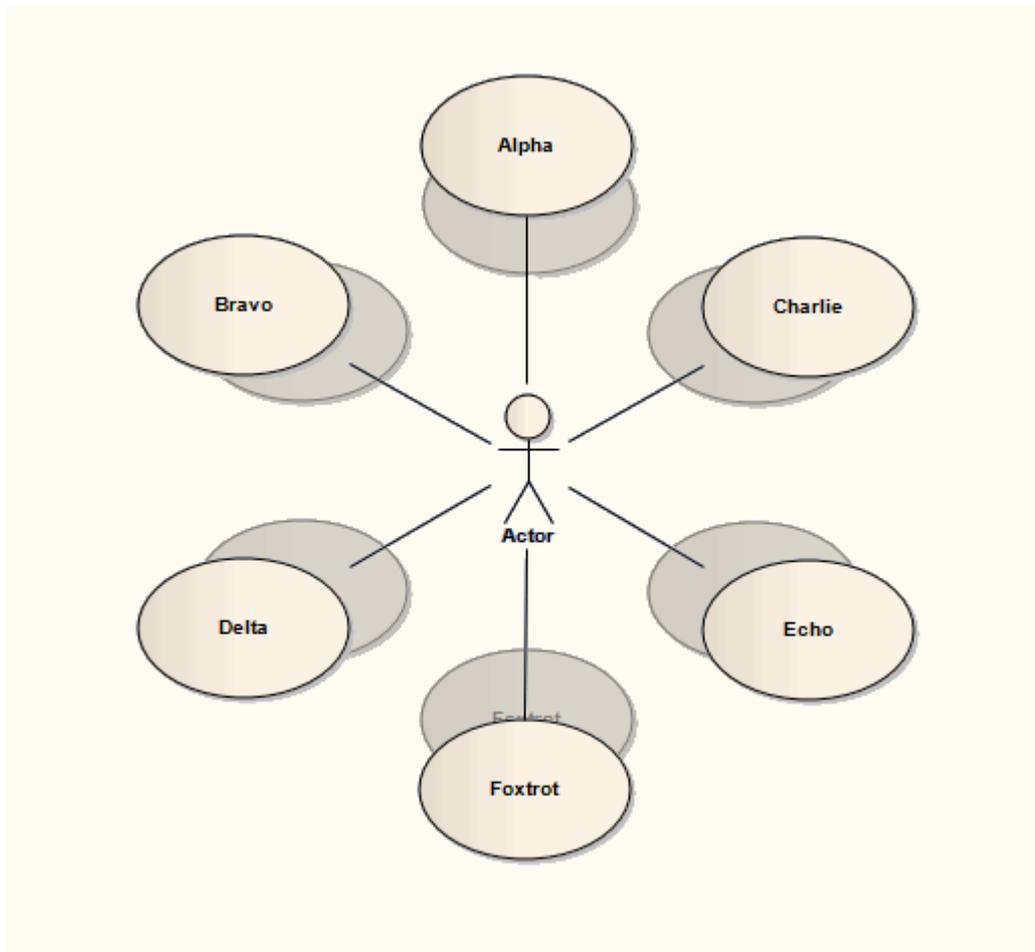
To invoke the Neaten layout, follow the steps below:

Step	Action	See Also
1	Follow the general Layout Diagrams procedure, and at step 3 select Neaten	Layout Diagrams ⁶⁰⁴
2	In the Threshold (px) field, type the height or width distance, in pixels, at which elements should be considered to be in the same row or column; a lower threshold value produces a tighter result, with only elements that are extremely similar - vertically or horizontally - considered to be in the same row or column	
3	In the Column Snap field, click on the drop-down arrow and select the appropriate alignment for elements in the same column:	

Step	Action	See Also
	<ul style="list-style-type: none">• Left - elements are aligned with the left edge of the left-most element in the column• Center - elements are aligned with the vertical center of the center-most element in the column• Right - elements are aligned with the right edge of the right-most element in the column	
4	<p>In the Row Snap field, click on the drop-down arrow and select the appropriate alignment for elements in the same row:</p> <ul style="list-style-type: none">• Top - elements are aligned with the top edge of the highest element in the row• Center - elements are aligned with the horizontal center of the center-most element in the row• Bottom - elements are aligned with the bottom edge of the lowest element in the column	

5.5.7.7 Converge/Diverge Layout

The *Converge* layout attracts the set of selected elements towards the center of their bounding rectangle. Conversely, the *Diverge* layout repels the set of selected elements away from the center of their bounding rectangle. The Converge/Diverge layout also tries to maintain connector angles if an element in the set contains a connector with waypoints.

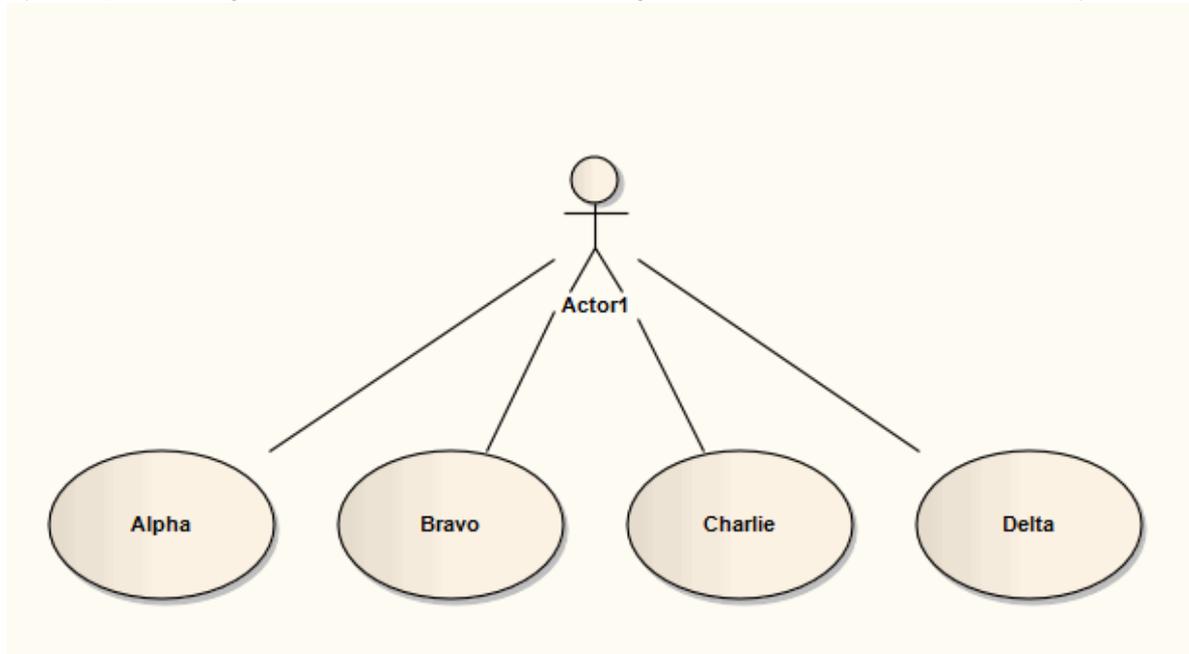
**How to:**

To invoke the Converge/Diverge layout, follow the steps below:

Step	Action	See Also
1	Follow the general Layout Diagrams procedure, and at step 3 select Converge/Diverge	Layout Diagrams [604]
2	For Direction , select the required layout: <ul style="list-style-type: none"> • Converge - attracts the set of selected elements to the center point • Diverge - repels the set of selected elements from the center point 	
3	The Amount (%) slider determines how far the elements are moved towards or away from the center point; the movement is the element's current distance from the center point multiplied by the percentage value set on the slider In the Converge layout, the element moves towards the center point; in the Diverge layout the element moves further away from the center point Set the slider to the required percentage	

5.5.7.8 Fan Relations Layout

The *Fan Relations* layout arranges the immediate relations of an element around a specified edge. This layout requires a single element to be selected on the diagram, to be used as the context for the layout.



How to:

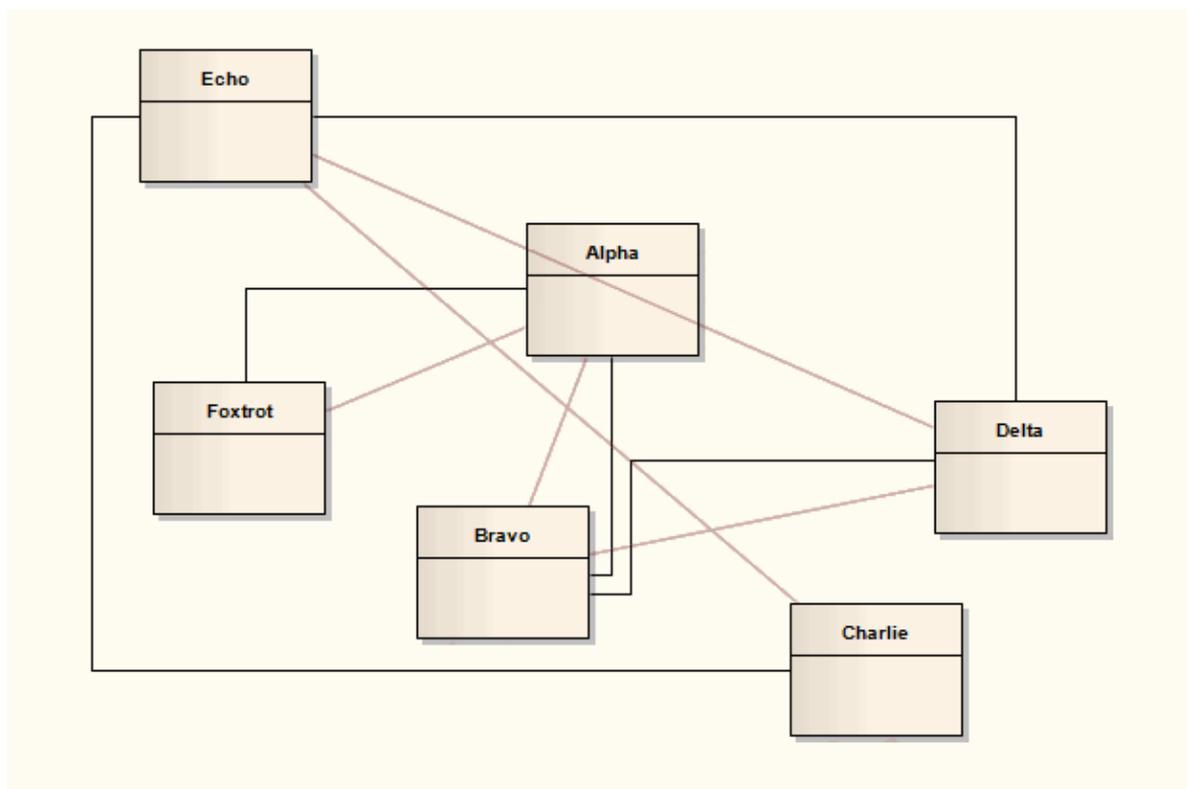
To invoke the Fan Relations layout, follow the steps below:

Step	Action	See Also
1	Open the Layout Tools window; either: <ul style="list-style-type: none"> Select the View Diagram Layout menu option, or Right-click on the main toolbar and select the Layout Tools context menu option 	Layout Diagrams 604
2	Select the single element around which to lay out related elements on the currently-active diagram	
3	Click on the drop-down arrow on the top left button of the Layout Tools window, and select Fan Relations	
4	Click on the drop-down arrow in the Sort By: field and select the required sort parameter The options are: <ul style="list-style-type: none"> None - Elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom) Area (Ascending) - Elements are passed to the specified layout in order of the screen space they occupy, smallest to largest Area (Descending) - Elements are passed to the specified layout in order of the screen space they occupy, largest to smallest Name (Ascending) - Elements are passed to the specified layout in alphanumeric order, based on the element name Name (Descending) - Elements are passed to the specified layout in reverse alphanumeric order, based on the element name 	

Step	Action	See Also
	<ul style="list-style-type: none"> • Element Type - Elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name 	
5	In the Padding (px) field, type the separation required between the selected element and its related elements, in pixels	
6	Select the connector direction to use in determining the related elements to lay out; either: <ul style="list-style-type: none"> • Incoming Nodes - to lay out related elements that have the selected element as the target • Outgoing Nodes - to lay out related elements that have the selected element as the source 	
7	In the Fan Edge field, click on the drop-down arrow and specify the edge of the selected element from which to lay out the related elements <ul style="list-style-type: none"> • Left - to arrange related elements to the left of the selected element • Right - to arrange related elements to the right of the selected element • Top - to arrange related elements from the top of the selected element • Bottom - to arrange related elements from the bottom of the selected element 	

5.5.7.9 Auto Route Layout

The *Auto Route* layout orthogonally routes connectors between the selected elements. The layout attempts to find the shortest path between the two connected elements while minimizing crossings. In the following layout, the original connectors are shown in red.



How to:

To invoke the Autoroute layout, follow the steps below:

Step	Action	See Also
1	Follow the general Layout Diagrams procedure, and at step 3 select Auto Route	Layout Diagrams ^[604]
2	When calculating connector routes, the algorithm divides the diagram into cells of a size determined by the Cell Size value; a smaller cell size results in connectors being placed closer together In this Cell Size (px) field, type the value in pixels	
3	In the Element Margin field, type the preferred separation between connector segments and element borders, in pixels	

5.5.7.10 Lay Out a Diagram Automatically

Enterprise Architect provides the facility to layout diagrams automatically. This creates a tree-based structure from the diagram elements and relationships in a diagram. Owing to the complexity of many diagrams, you might then have to do some manual 'tweaking'.

How to:

To layout a diagram, follow the steps below:

Step	Action	See Also
1	Select a diagram.	
2	Click on either: <ul style="list-style-type: none"> The Diagram Layout Diagram option, or The Auto Layout button on the diagram toolbar 	Layout Diagrams ^[604]

Notes:

- This facility is available for Structural diagrams and Extended diagrams, but not for Behavioral diagrams (see the *UML Diagrams* topic for a description of the diagram types); however, the facility is also available for Sequence diagrams generated by the Enterprise Architect Debugger
- Dynamic and Analysis diagrams are **NOT** suited to this form of layout - please ensure first that the diagram type you are laying out benefits from the action
- If you dislike the autolayout, you can reverse it before saving the diagram; click (**Ctrl+Z**)

For a fine degree of control of the elements in your diagram, you can use the Diagram Layout Options dialog. Generally the default layout parameters provide adequate layouts for a wide range of diagrams, but there are times when more specific settings are required.

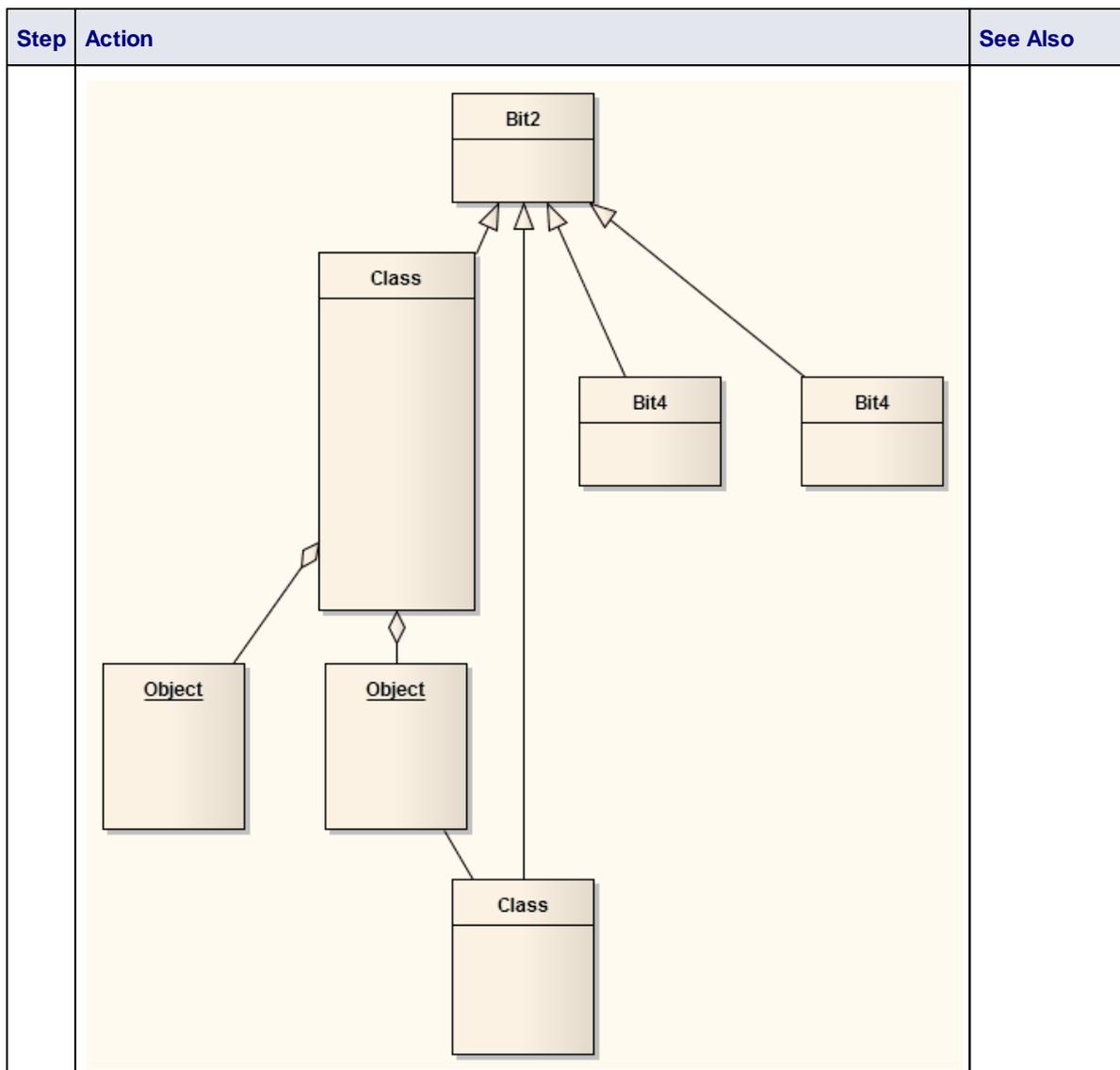
How to:

To access the Diagram Layout Options dialog, follow the steps below:

Step	Action	See Also
1	Double-click on the background of the diagram	

Step	Action	See Also
	The Diagram Properties dialog displays	
2	Click on the Diagram tab, then click on the Set Layout Style button The Diagram Layout Options dialog displays	
3	<p>When you have made the required changes, click on the OK button to save the changes</p> <p>You can alter any of the following settings on the Diagram Layout Options dialog to refine your layout:</p> <ul style="list-style-type: none"> • Cycle Remove Options panel - these settings remove cycles in the element organization (where element X is the source of a path but also becomes the target of a branch of the path), by reversing the connectors that impose the cycling and then reorganizing the diagram and reinstating the reversed relationships; this identifies the primary source element in the diagram <ul style="list-style-type: none"> • Greedy - Select to use the Greedy Cycle Removal algorithm, which minimizes the number of connectors reversed • Depth First Search - Select to use the Depth First Search Cycle Removal algorithm, which establishes the longest linear sequence possible, before establishing parallel sequences and branches; this algorithm is less effective in large and/or complex diagrams, but produces a more natural layout than the Greedy algorithm • Crossing Reduction Options panel - these options determine how long the routine should look for ways of reorganizing the layout to avoid crossed relationships: <ul style="list-style-type: none"> • Iterations - Type the number of iterations to be used during cycle removal (more than 8 does not usually provide any improvement) • Aggressive - Select to use an aggressive (detailed and time-consuming) crossing reduction step • Layering Options panel - these settings determine how elements are organized in layers during layout: <ul style="list-style-type: none"> • Longest Path Sink - Select to use the Longest Path Sink Layering algorithm, where the final target elements (sinks, which have no relationships issuing from them) are arranged in a layer at the top of the diagram, and the relationship paths built downwards from there in as many layers as there are nodes in the longest path • Longest Path Source - Select to use the Longest Path Source Layering algorithm, where the original source elements (those with no relationships entering them) are arranged in a layer at the bottom of the diagram and the relationship paths built up from there in as many layers as there are nodes in the longest path • Optimal Link Length - Select to use the Optimal Link Length Layering algorithm, which organizes the elements into whichever layers minimize the total source-to-sink relationship chain; in this layout you can have both source elements and sink elements at various levels of the diagram • Layout Options panel: <ul style="list-style-type: none"> • Layer Spacing - Type the default number of logical units between layers of elements (vertical spacing) • Column Spacing - Type the default number of logical units between elements within a layer (horizontal spacing) • Direction panel: <ul style="list-style-type: none"> • Up, Down, Left, Right - Select the direction in which directed connectors should point, to set the position of the primary source element and the overall flow of the diagram • Initialize Options panel - the autolayout routine inserts line waypoints and 	

Step	Action	See Also
	<p>connectors into relationship paths to help plot the direction of relationships; the routine then assigns an index number to every node, such that nodes in the same layer are numbered left to right</p> <p>The settings in this panel determine how those index numbers are assigned</p> <ul style="list-style-type: none"> • Naive - Select to use the Naive Initialize Indices algorithm, which assigns index numbers to nodes as they are encountered in a sweep and tends to place all waypoints to the right of real nodes (and therefore long relationships between a small number of elements to the right of chains of short relationships between several elements) • Depth First Search Outward - Select to use the Depth First Out Initialize Indices algorithm, which assigns index numbers to nodes as they are encountered in a depth first search from source nodes outwards (and would therefore place longer relationship chains to the left of shorter chains, with the primary source node at the start of the diagram flow) • Depth First Search Inward - Select to use the Depth First In Initialize Indices algorithm, which also assigns index numbers to nodes as they are encountered in a depth first search, but from sink nodes inwards (and would therefore place longer relationship chains to the left of shorter chains, with the ultimate target node at the end of the diagram flow) <ul style="list-style-type: none"> • Set as Project Default checkbox - select this checkbox to apply the diagram layout settings to all diagrams in the project; if you later check this box and click on the OK button for a different diagram, the new settings override the settings saved earlier <p>The following is an example of an automatically laid out diagram, with the following options set:</p> <ul style="list-style-type: none"> • Depth First Search • Optimal Link Length • Depth First Search Outward • Direction - Up 	

**Learn More:**

- [UML Diagrams](#)^[795]

5.5.8 The Quick Linker

The *Quick Linker* provides a simple and fast way to create new elements and connectors on a diagram. When an element is selected in a diagram, the **Quick Linker** arrow is displayed at the upper right corner of the element, as shown below:

Topics:

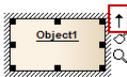
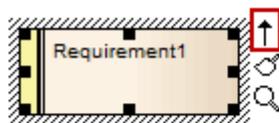
Image	Topic	Detail	See also
	Usage	Simply clicking and dragging the icon enables you to create new connectors and elements on a diagram, as explained in the following topics: <ul style="list-style-type: none"> • Create New Elements 	Create New Elements ^[623] Create Connectors ^[624]

Image	Topic	Detail	See also
		<ul style="list-style-type: none"> • Create Connectors <p>The connectors and elements suggested by the Quick Linker are the commonest objects appropriate to the context. You can select others from the Toolbox pages. Also, a Technology Developer can edit the lists of elements and connectors, and create new combinations.</p>	

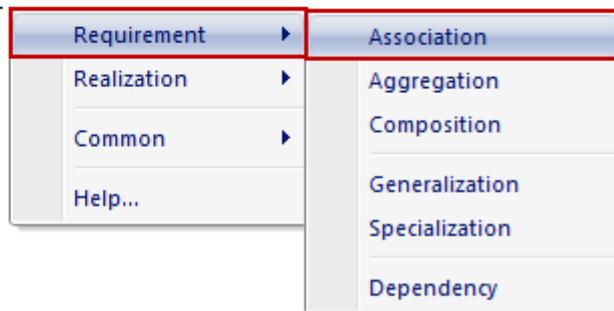
5.5.8.1 Create New Elements



Click, drag and release



Choose element and connector type



Topics:

Topic	Detail	See also
Usage	<ul style="list-style-type: none"> • Press and hold (Shift) while selecting the type of connector to select an existing classifier as the target. • For rapid modeling, you can suppress the Properties dialog when creating new elements. See the option Tools Options Objects Edit Object on New. <p>The available Quick Linker options depend on the type of element selected. For example, the Quick Linker options for a Class (above) differ from those of an Actor (below). These are the most appropriate, commonly used elements and connectors for the source element; you can create other target elements and connectors by selecting them from the appropriate Toolbox page.</p>	

5.5.8.2 Create Connectors



Click, drag and release



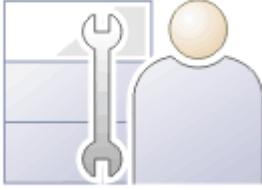
Choose element and connector type



Topics:

Topic	Detail	See also
Usage	<p>The list of connectors provides the most appropriate, commonly-used connectors for the source and target element types. If you want to use a different connector, select the appropriate Toolbox page, click on the required connector and then on the source element, and drag across to the target element.</p> <p>The connector does not actually establish until you release the mouse button over the target element. However, a dotted line shows where the connector would be at any point, and the solid outline of the nearest element or extension changes to a hatched outline as you move the cursor onto it; this helps you identify where the connector should connect to, if there are many closely-arranged elements, Parts, Ports and other extensions.</p> <p>You can also bend the connector, pressing (Shift) as you drag the cursor in a new direction.</p> <p>You can create a Notelink to attach a Note or Constraint to a connector, simply by dragging the Quick Linker arrow on the Note to the connector.</p>	

5.6 Elements



Models are constructed from elements, each of which has its own meaning, rules and notation. Elements can be used at different stages of the design process for different purposes.

The properties and features of an element can be defined and displayed through a set of windows and dialogs.

Learn More:

- [Element Property Displays](#) ^[660]
- [Element Tasks](#) ^[625]
- [Element Context Menu](#) ^[648]
- [Visual Representation](#) ^[658]
- [In-place Editing Options](#) ^[720]

5.6.1 Element Tasks

This topic describes the following common tasks that you can perform on elements in Enterprise Architect:

Topic	Link
Create Elements	Create Elements ^[626]
Add Elements Directly to Packages	Add Elements Directly to Packages ^[627]
Use Auto Naming and Auto Counters	Use Auto Naming and Auto Counters ^[628]
Set Element Parent	Set Element Parent ^[628]
Show Element Use	Show Element Use ^[630]
Set Up Cross References	Set Up Cross References ^[630]
Move Elements Between Packages	Move Elements Between Packages ^[633]
Move Elements Within Diagrams	Move Elements Within Diagrams ^[632]
Copy Elements Between Packages	Copy Elements Between Packages ^[634]
Change Element Type	Change Element Type ^[635]
Align Elements	Align Elements ^[635]
Resize Elements	Resize Elements ^[636]
Delete Elements	Delete Elements ^[637]
Customize Visibility of Elements	Customize Visibility of Elements ^[638]
Create Notes and Text	Create Notes and Text ^[639]
Link Note to Internal Documentation	Link Note to Internal Documentation ^[640]
Set an Element's Default Appearance	Set an Element's Default Appearance ^[641]
Use Element Templates	Use Element Templates ^[643]
Highlight Context Element	Highlight Context Element ^[645]
Make Linked Element a Local Copy	Make Linked Element a Local Copy ^[646]

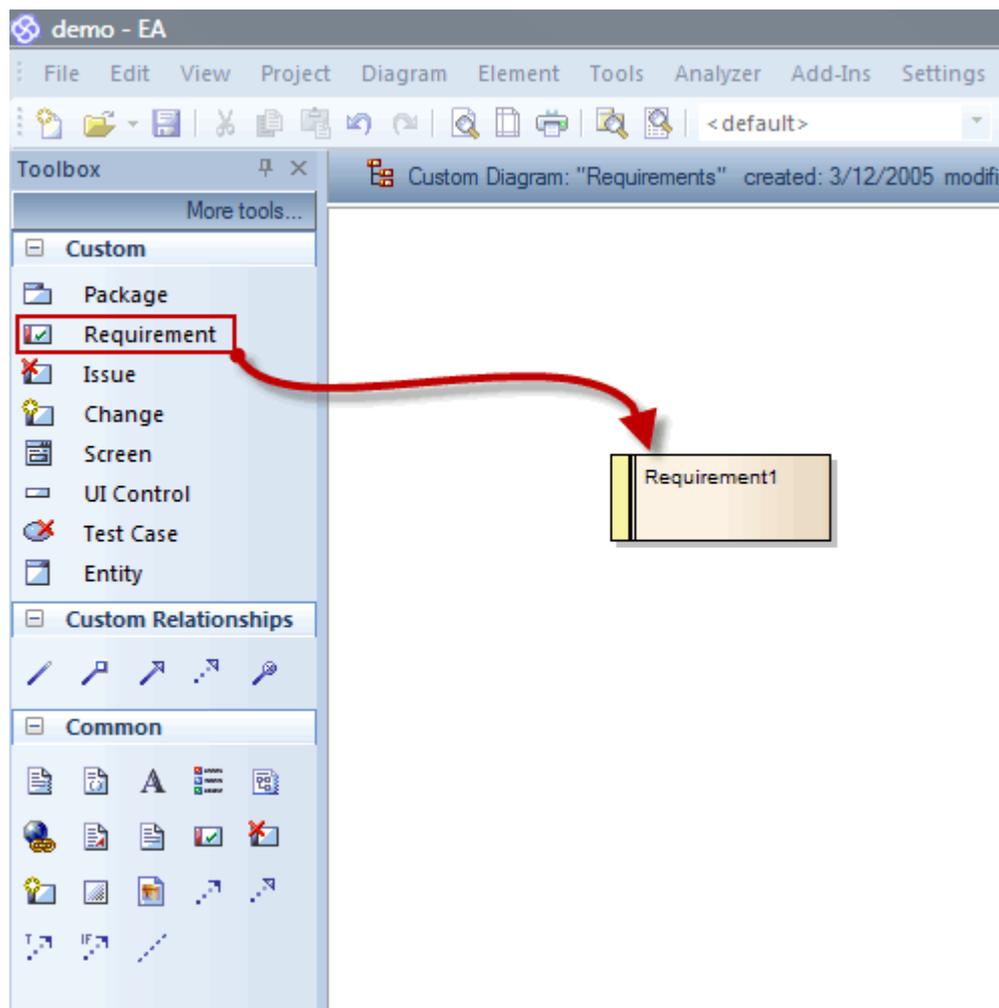
Topic	Link
Copy Features (Attributes and Operations) Between Elements	Copy Features (Attributes and Operations) Between Elements ^[646]
Move Features Between Elements	Move Features Between Elements ^[647]

Notes:

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Element permission to update element properties or delete an element

Learn More:

- [Permissions List](#)^[205]

5.6.1.1 Create Elements**Example:****Notes:**

- If you have defined a template element in the Templates package, Enterprise Architect applies the characteristics of that template to each new element you create of that type
- If you are creating several elements of one type, after creating the first just press **(Shift+F3)** or **(Ctrl) + click** to create the next element of that type
- Be aware that once you have created elements, you can re-use them by dragging them from the Project Browser and dropping them onto your diagrams
- Sometimes it is useful to add elements to a package, without a diagrammatic representation; this can be accomplished via the Project Browser and is explained in the topic - *Add Elements Directly to a Package*

Learn More:

- [Create Elements In Place Using the Quick Linker](#)^[623]
- [Create Elements Using the Toolbox](#)^[546]
- [Create Elements Using the Diagram Context Menu](#)^[538]
- [Create Elements From Text File](#)^[1156]
- [Create a Group of Elements Using UML Patterns](#)^[1021]
- [Create Domain Specific Elements from UML Profiles](#)^[1026]
- [Pasting from the Tree](#)^[576]
- [Add Elements Directly to a Package](#)^[627]
- [Set Element Templates Package](#)^[643]
- [Behavioral Diagram Elements](#)^[864]
- [Structural Diagram Elements](#)^[940]

5.6.1.2 Add Elements Directly To Packages

You can quickly add new elements to a package without the necessity of adding a diagram element at the same time. This is particularly useful in defining a group of Requirements, Changes, Issues, base Classes or other element types that might not require diagrammatic representation in the model.

How to:

To add a new element to a package, follow the steps below:

Step	Action	See Also
1	In the Project Browser , right-click on the appropriate package. The context menu displays.	
2	Select the Add Add Element menu option. The New Element dialog displays.	
3	In the Name field, type the name of the element.	
4	In the Type field, click on the drop-down arrow and select the element type.	
5	If required, in the Stereotype field either type the stereotype name or click on the drop-down arrow and select the stereotype.	
6	Select the Open Properties Dialog on Creation checkbox if the Properties dialog is to open immediately after the element is created.	
7	Deselect the Close Dialog on OK checkbox to add multiple elements in one session.	
8	Click on the OK button to create the element.	

Notes:

- The drop-down list is populated from one of the **Toolbox** page groups (including profile, Add-In and MDG Technology groups). If the list does not represent the group containing the element you require, click on the **Select Group** button and, from the list, select the appropriate **Toolbox** page group. The drop-down list then shows the elements from that group
- The <default> group in the list contains a basic set of elements drawn from across the UML Behavioral and Structural groups, and the Enterprise Architect Extended groups
- If you have a diagram open, the **Add to Current Diagram** checkbox is available and defaulted to selected to add the new element to the diagram. If you do not want the element in that diagram, deselect the checkbox

5.6.1.3 Use Auto Naming and Auto Counters

The Auto Element Naming dialog enables you to configure automatic naming for any element type. Each element can have separately configured automatic names and aliases.

How to:

To set up auto naming, follow the steps below:

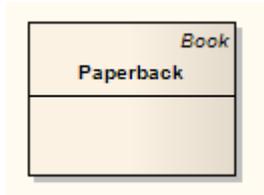
Step	Action	See Also
1	<p>In the Name panel:</p> <ul style="list-style-type: none"> • In the Prefix field, type a prefix for the new name (optional) • In Counter field, type the counter value; use as many 0's as required to pad the name • In the Suffix field, type a suffix for the new name (optional) • If required, click on the Active checkbox to turn auto naming on for this element type 	
2	<p>In the Alias panel:</p> <ul style="list-style-type: none"> • In the Prefix field, type a prefix for the new alias (optional) • In Counter field, type the counter value; use as many 0's as required to pad the alias • In the Suffix field, type a suffix for the new alias (optional) • If required, click on the Active checkbox to turn alias auto naming on for this element type 	
3	<p>Click on the Save button.</p> <p>New elements of this type now have an automatically-generated name and/or alias with an incrementing counter value.</p> <p>If an Alias is active then auto naming applies; however, to view the Alias in a diagram requires that the option Use Alias if Available is selected in Diagram Properties.</p>	Diagram Properties 

5.6.1.4 Set Element Parent

You can manually set an element's parent or parents, or an interface it realizes (implements), using the **Set Parents and Interfaces** dialog.

How to:

To set an element parent, follow the steps below:

Step	Action	See Also
1	Select a generalizable element in a diagram.	
2	Select the Element Advanced Set Parents and Interfaces menu option. Alternatively: <ul style="list-style-type: none"> • Press (Ctrl+I) or • Right-click and select the Advanced Parent context menu option. The Set Parents and Interfaces dialog displays.	
3	You can elect to enter a parent or interface name by either manually typing it in, or clicking on the Choose button to locate the element within the current model.	
4	Set the Type of relationship (Implements or Generalizes) from the drop-down list.	
5	Click on the Add button to add the relationship to the Type Details list, and to the selected child element. 	
6	Repeat this process to add further parent elements, or click on the Delete Selected button to remove a selected relationship.	

If the parent name is shown in the element on the diagram, you can select it and perform further operations, as follows:

Step	Action	See Also
1	Click on the element	
2	Click on the parent name to highlight it	
3	Right-click to display the context menu, which provides options enabling you to: <ul style="list-style-type: none"> • Redisplay the Set Parents and Interfaces dialog to, for example, delete the parent or add further parents • Display the parent element Properties dialog • Find the parent element in the Project Browser Find the parent element in the diagram If the parent element does not currently exist in the model, and is listed on the element in the diagram, you can highlight it and click on it to redisplay the Set Parents and Interfaces dialog to, for example, delete it or add further parents	

Notes:

- You can specify parent elements that exist in the model, or that you intend to create later. If the parent element does not currently exist in the model, select the **Accept classifier even if not in model** checkbox to ensure that the reference to a non-existent element is accepted
- If Parents are not in the same diagram as their corresponding related element, the parentage is

shown in the top right corner of the child element

5.6.1.5 Show Element Use

You can display the use of an element using the **Element Usage** dialog. This lists all occurrences of the element throughout the model, and enables you to easily navigate to any occurrence.

How To:

To show element usage, follow the steps below:

Step	Action	See Also
1	Select an element in a diagram	
2	Select the Element Find in Diagrams menu option (or, for a Package element, the Find in all Diagrams option) Alternatively, press (Ctrl+U) If the element exists in other diagrams, the Element Usage dialog displays, listing all occurrences of the current element in diagrams in the model	
3	If you want to display the usage information in a more readable layout, you can resize the dialog and its columns Either: <ul style="list-style-type: none"> • Double-click on a line item to open the relevant diagram and display the selected element (the Element Usage dialog remains open), or • Click on the Open button to display the selected diagram and close the Element Usage dialog 	

Notes:

- You can also access this feature from the Project Browser; select an element in the tree and select the **Element | Find in Diagrams** menu option

If there is only one instance of the element in any diagram, that diagram displays instead

5.6.1.6 Set Up Cross References

It is possible to set up a cross reference (or Custom Reference) from one element in Enterprise Architect to another. You can also view existing cross references on an element, using the Context References tab on the element's Properties dialog, or the Traceability window.

How to:

To set up a cross reference, follow the steps below:

Step	Action	See Also
1	In the Project Browser , locate the target element or diagram (that is, the object of the cross reference).	
2	Open a diagram that contains the elements that are to have the currently selected element as a reference.	
3	Right-click on the element in the Project Browser . The context menu displays.	

Step	Action	See Also
4	Select the Add custom reference menu option.	
5	In the Set up references dialog, select the checkbox against each element to that is to have the target element as a reference.	
6	Optionally, in the Comment field, type some text to describe the purpose of the reference.	
7	Click on the OK button.	

To use the cross reference, follow the steps below:

Step	Action	See Also
1	Select an element in a diagram.	
2	Select the Element Advanced Custom References menu option Alternatively, either press (Ctrl+J) , or right-click on the element and select the Find Custom References context menu option.	
3	The Custom References dialog displays, showing a list of elements that have been set as cross references for the selected element.	
4	You can open the Properties dialog for an element by highlighting it and clicking on the Open button.	
5	If you have a diagram cross reference, you can open that diagram.	
6	If you have a string of diagram links, click on the Home button to return to the original diagram.	

Notes:

- You can delete a cross reference by selecting it on the Custom References dialog and clicking on the **Delete** button; cross references are also automatically deleted if the source or target element in the reference is deleted

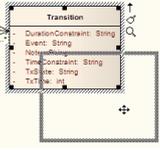
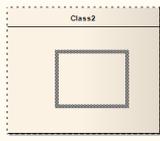
Learn More:

- [Context References Tab](#)^[686]
- [The Traceability Window](#)^[495]

5.6.1.7 Move Elements Within Diagrams

Any one of the following options enables you to move an element within a diagram.

Topics:

Image	Topic	Detail	See also
	Usage	<p>Select an element or group of elements in the diagram view, then:</p> <ul style="list-style-type: none"> • Use the mouse to drag the element to the required position (the cursor switches to the four-arrow icon as shown) • Hold down (Shift) and use the arrow keys to move the element by increments to the required position • Use the Left, Right, Up and Down options in the Element Move submenu • Align multiple elements using the Element Alignment submenu, the Alignment options in the right-click context menu, or the Alignment buttons on the Diagram toolbar. 	
	Confirm Possible Parent Elements	<p>As you organize the elements within a diagram, you can drag any element over another and, provided the dragged element is within and on top of the possible parent, it is always encapsulated by the lower element and moved within the lower element. However, the lower element might not be a valid parent.</p> <p>You can confirm that a possible parent element is able to accept a selected child element. When you drag the child element over the potential parent, the target element border changes to a dashed line if it can accept the selected element as a child. If the border does not change, the selected element cannot be a child to the target element.</p>	

Notes:

- The **Support for Composite Objects** checkbox must be selected on the Objects page of the Options dialog (select the **Tools | Options | Objects** option). If this option is not selected, the dashed border does not show and the child element cannot be embedded on the parent in the diagram
- Both elements must already exist on the diagram; an element border does not change if you drag a potential child element over it from the Toolbox or Project Browser
- The child element must have equal or higher Z-order placement than the parent; that is, the parent element must be level with or behind the child
- The child element borders must be completely within the parent element borders
- For example, if you drag a Signal over a Class, the Class border changes; a Class element can be a parent to a Signal. If you drag a Class element over a Signal element, the Signal border does not change. A Signal cannot be a parent to a Class
- When you embed a child element on its parent, the child element becomes part of the parent element hierarchy in the Project Browser. Similarly, if you drag the child element out of the parent, the child element becomes independent and is no longer embedded in the parent element hierarchy

Learn More:

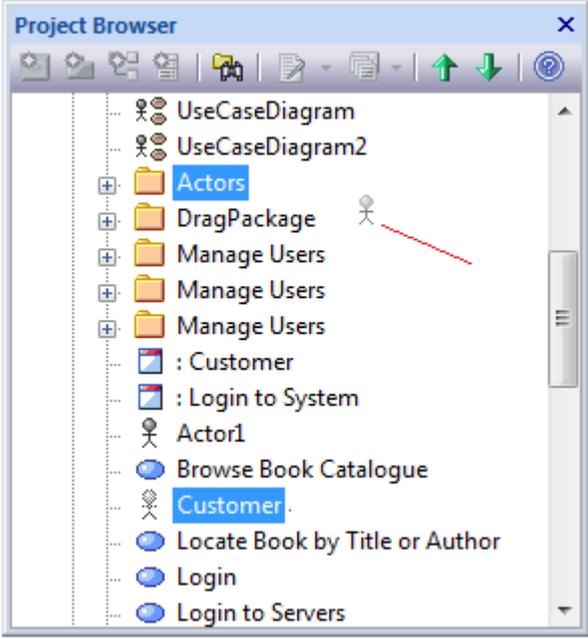
- [Objects](#)^[432]
- [Diagram Context Menu](#)^[538]

5.6.1.8 Move Elements Between Packages

Elements and packages can be moved from one package to another by dragging and dropping the element to the target destination in the **Project Browser**. Note that if you move a package, ALL the child packages and their contents are moved to the new location also.

How To:

To move an element between packages, follow the steps below:

Step	Action	See Also
1	Click on the element in the Project Browser. (See Customer in the diagram below.)	
2	<p>Drag the element so that the cursor is over the target package icon. The element symbol (and, in some operating systems, the element name) displays at the moving cursor position.)</p> 	
3	Release the mouse button. The element is moved into the target package.	

Notes:

- You can also drag the element under a host element in the new package; for example, drag an element under a Class
- Moving an element has no effect on any relationships that the element might have
- Moving an element in the **Project Browser** has no effect on the use of that element in any diagram
- Moving a diagram generally does not affect the location of elements in packages. If you move a diagram out of one package into another, all the elements in the diagram remain in the original package. However, certain elements (such as Decision, Initial and Final elements) are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram. Therefore, if you move a diagram containing these elements, they **are** moved to the new parent package with the diagram

- In a multi-user environment, if one person moves or updates the **Project Browser** structure, other users must reload their project to see the latest changes in the **Project Browser**. Although this is true of any addition or modification to the tree, it is most important when big changes are made, such as dragging a package to a different location

Learn More:

- [Refresh View of Shared Project](#)^[187]

5.6.1.9 Copy Elements Between Packages

Enterprise Architect enables you to quickly and easily duplicate one or more elements, including their child elements and diagrams. You can insert a copy of an element under one or more other packages, in the same .eap file or any other .eap file.

Topics:

Topic	Detail	See also
Copying Elements	<p>In the Project Browser, select each required element, right-click on one of them and select the Copy Element(s) to Clipboard context menu option (or click on a selected element and press (Ctrl+C))</p> <p>The Copy Element(s) to Clipboard dialog briefly displays until the copy operation completes</p>	
Pasting Elements	<p>In the Project Browser, right-click on the package into which to paste the copied elements, and select the Paste Element(s) from Clipboard context menu option (or click on the package and press (Ctrl+V))</p> <p>The Paste Element(s) from Clipboard dialog briefly displays until the paste operation completes.</p> <p>The target package is expanded and the pasted elements are exposed in the Project Browser. If you are pasting the elements within the same model as the copied source, the source parent package is also collapsed.</p> <p>If the target package already contains:</p> <ul style="list-style-type: none"> • an element of the same type with the same name as a pasted element, the pasted element name has the suffix - Copy • an element with the same name as the pasted element including the - Copy suffix, the suffix becomes - Copy1 (or - Copy 2, - Copy3 and so on, as copies of the element accumulate in the target package) <p>You can keep the same element names as the source, or you can rename each element either by clicking twice on it and editing the name in the Project Browser, or by double-clicking on it and editing the name in the Properties dialog</p>	

Notes:

- A copy of an element does not have the external cross references of the source element; that is:
- if one element is copied it has no connectors
- if more than one element is copied, only the connectors between the copied elements are retained
- however, if those elements come from a Sequence or Communication diagram and the diagram itself

is not copied, the message connectors between the copied elements are not retained

- You cannot paste an element into a package that is locked by another user or that is checked in. The **Paste...** option is grayed out in the context menu

Learn More:

- [Locking Packages](#) [212]
- [Checking In/Checking Out Packages](#) [275]

5.6.1.10 Change Element Type

Access: **Element | Advanced | Change Type**

Select the new element type from the list and click on the **OK** button.

The target element is transformed into the required type.

5.6.1.11 Align Elements

To align multiple elements, follow the steps below

How To:

Step	Action	See Also
1	Select a group of elements by drawing a selection box around them all (or select them one by one by holding down (Ctrl) and clicking on each element)	
2	Right-click on the element in the group to align others to The context menu displays	
3	Select the alignment function you require All selected elements are aligned to the one beneath the cursor	

Notes:

- You can also use the Diagram toolbar; the first four buttons are used to align elements, and are made available when more than one element is selected in a diagram
- You can also select the **Element | Alignment** menu option, to display one of the Position submenus

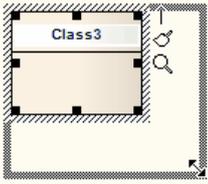
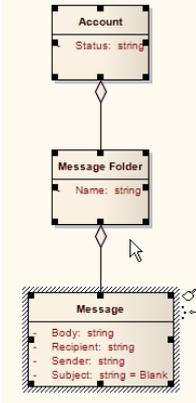
Learn More:

- [Diagram Toolbar](#) [111]
- [Position Submenus](#) [88]

5.6.1.12 Resize Elements

Any one of the following options enables you to resize an element.

Topics:

Images	Topic	Detail	See also
	Resize handles	<p>Select an element or group of elements in the diagram view, then:</p> <ul style="list-style-type: none"> • Use the resize handles that appear at each corner and side to resize the element(s) by dragging with the mouse (the cursor switches to the double-ended arrow as shown in the image to the left) • Press and hold (Ctrl) and use the arrow keys to resize by increments as required • Use the Wider, Narrower, Taller and Shorter options in the Element Move submenu • Autosize selected element(s) using the option in the Element Appearance submenu, or by pressing (Alt+Z). (With multiple elements selected, Autosize also appears in the right-click context menu) • Set multiple elements to the same height, width or both, using these options in the Element Make Same submenu, or the options in the right-click context menu. 	
	Resize a Set of Objects to a specific size	<p>If you right-click a selected set of objects, you can resize them to the same dimensions (height, width or both) using the context menu. When you select multiple elements using (Ctrl) +click, then resize the dimensions, the dimensions of the selected hatched object are used to set the dimensions of the other selected objects.</p> <p>For example, in the diagram below, the Message Class height and width are used to set the height and width of the Account and Message Folder Classes. The aim is to make the Account and Message Folder elements the same height and width as the Message element.</p>	

How To:

To resize a set of objects to a specific size, follow the steps below:

Step	Action	See Also
1	Set one element to the required size (for example, Message as above)	
2	Select all other elements (for example, Account and Message Folder as above)	

Step	Action	See Also
3	Right-click on the pre-sized element (for example, Message)	
4	Select your resizing option (such as same height, width) from the context menu	

Learn More:

- [Highlight Context Elements](#) ⁶⁴⁵

5.6.1.13 Delete Elements

How To:

To delete an element from a diagram, follow the steps below:

Step	Action	See Also
1	In the active diagram, click on the element to delete	
2	Either: <ul style="list-style-type: none"> • Press (Delete) or • Right-click to display the context menu and select the Delete <element name> option This does not delete the element from the model, only from the current diagram	

To delete an element from the model, follow the steps below:

Step	Action	See Also
1	In the Project Browser, right-click on the element to delete The context menu displays	
2	Select the Delete <element name> option A confirmation prompt displays	
3	Click on the Yes button	

Alternatively, click on the element in the Project Browser and press (**Ctrl+Delete**). The element is completely removed from the **model**.

To delete multiple elements from a diagram, follow the steps below:

Step	Action	See Also
1	In the active diagram, (Ctrl) +click on each element to delete	

Step	Action	See Also
2	Either: <ul style="list-style-type: none"> • Press (Delete), or • Right-click to display the context menu and select the Delete selected elements option 	

To delete multiple elements from a diagram and model, follow the steps below:

Step	Action	See Also
1	Open the diagram containing the elements to remove from the model	
2	Press (Ctrl+A) to select all of the elements in the diagram, or use (Ctrl) + click to select specific elements	
3	Press (Ctrl+Delete) to completely remove the elements from the model	

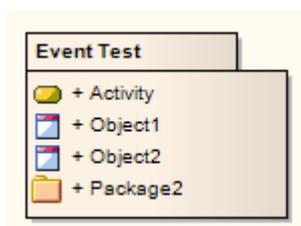
To delete multiple elements from the Project Browser and model, follow the steps below:

Step	Action	See Also
1	In the Project Browser, press and hold either (Shift) or (Ctrl) and click on the required items	
2	To completely remove the elements from the model, either: <ul style="list-style-type: none"> • Press (Ctrl+Delete), or • Right-click on the selected items and select the Delete selected item(s) context menu option If you delete an element in this way, you delete all its properties and connectors as well	

5.6.1.14 Customize Visibility of Elements

Some elements are hidden from view in packages and in RTF documents by default. These include Events, Decisions, Sequence elements and Associations. You have the option of turning these elements back on.

For example, some Events and Decisions contained in a package do not appear in the package view, as in the example below.



How To:

Images	Step	Action	See Also
	1	Select the Tools Options Objects menu option The Objects page of the Options dialog displays	
	2	Click on the Advanced button The Advanced Settings dialog displays	
	3	Select the checkbox for each type of element to show in packages and in RTF documents	
	4	Click on the Close button on each dialog	
	5	Reload the current diagram if required The package from the example shows the Event and Decision elements it now contains (see left image)	Refreshing the view of a shared project ^[187]

5.6.1.15 Create Notes and Text

You can create both notes and text on a diagram; the two are slightly different.

How To:

To create a Note, follow the steps below:

Step	Action	See Also
1	<p>Drag the Note icon from the Common page of the Toolbox onto the diagram.</p> <ul style="list-style-type: none"> If you have the Edit Object On New checkbox deselected on the Objects page of the Options dialog, the Note element displays on your diagram; type your note text directly within the Note element If you have the checkbox selected, the Notes window displays; type your text in that window <p>If you want to display the Notes information in a more readable layout, you can resize the dialog</p> <p>You can format the text if necessary, using the Notes toolbar at the top of the window. When you have completed the text, click on the OK button to save it</p>	Objects ^[432] Notes ^[769] Notes Toolbar ^[770]

To create a Text element, follow the steps below:

Step	Action	See Also
1	<p>Drag the Text Element icon from the Common page of the Toolbox onto the diagram</p> <p>The Notes window displays</p>	Notes ^[769]

Step	Action	See Also
2	<p>Type your text in the window</p> <p>If you want to display the Notes information in more readable layout, you can resize the dialog</p> <p>You can format the text if necessary, using the Notes toolbar at the top of the window; when you have completed the text, click on the OK button to save it</p>	Notes Toolbar 

Notes:

- You can also create a note by clicking on the **New Note** icon (the text page) on the UML Elements toolbar and clicking on the diagram

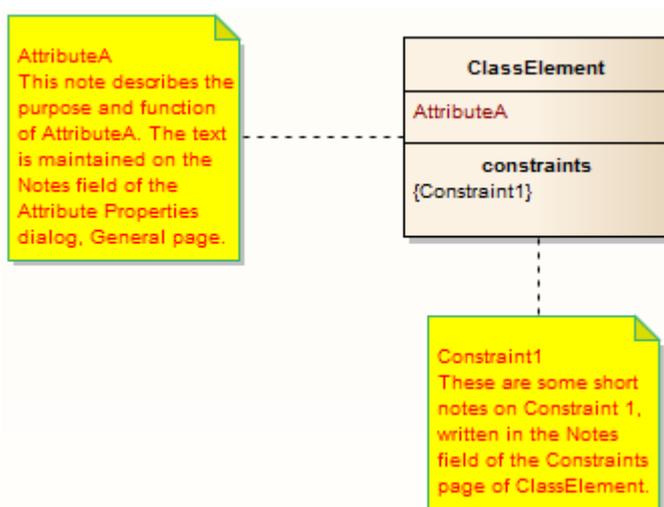
Learn More:

- [UML Elements Toolbar](#)

5.6.1.16 Link Note to Internal Documentation

It is possible to connect a *Note* element to another element's internal documentation. This enables you to externalize model documentation to the diagram level, and as Enterprise Architect keeps the note and the internal structure in synch, you do not have to worry about updating the note contents; this is done automatically.

In the example below, two notes are connected into an element's internal structures. One is connected to an attribute, and displays the attribute name and notes. The other is connected to a constraint, showing the constraint name and documentation.

**How To:**

To connect a Note element to another element's documentation, follow the steps below:

Step	Action	See Also
1	Click on the element and then click on the feature to link the Note to	
2	Select the Element Inline Features Create Linked Note context menu option. This creates a Note element linked to the selected feature, reflecting the content of that feature	

Alternatively:

Step	Action	See Also
1	Insert the target element into a diagram	
2	Drag the Note icon from the Common page of the Toolbox onto the diagram, next to the target element The Notes dialog displays	
3	Do not type any text, just click on the OK button	
4	Click on the Note Link icon from the Common page of the Toolbox	
5	Click on the Note, and drag across to the target element to create the connector	
6	Right-click on the Note Link to display the context menu	
7	Select the Link this Note to an Element Feature menu option The Link note to element feature dialog displays	
8	In the Feature Type field, click on the drop-down arrow and select the type of feature to link to	
9	In the Feature list, click on the specific feature to link to	
10	Click on the OK button The note now automatically derives its contents from the target element	

5.6.1.17 Set an Element's Default Appearance

To override the global appearance and define a default appearance for a specific element on all diagrams on which it is found, you use the Default Appearance dialog.

Access: **Double-click element | Appearance | Default Appearance**

Use to:

- Alter the background, border and font colors
- Change font styles, including font face and size
- Alter the border width for an element

Reference:

Field	Usage	See also
Background Color	Set the color of the background for an element	
Border Color	Set the color of the border for an element	
Border Width	Set the width of the border surrounding an element, the minimum setting being 1 , the maximum setting being 5	
Font Style (...)	Set the typeface of element text The current font typeface is displayed within the text field; click on the (...) button The Font dialog displays, which enables you to select a different typeface, style, (such as bold/italic) and font size You can also select from a limited number of font colors and effects on this dialog	Font Dialog <small>657</small>
Font Color	Set the font color of the element text	
Reset to defaults	Reset all appearance settings to default values	
Preview	Review a representation of the colors, border width and font styles, which updates as you choose the new settings	

Notes:

- You can adjust several elements at the same time; select all of the required elements, right-click on one of them and select the **Default Appearance** context menu option, or use the Format toolbar
- If the **Show Project Custom Colors in Color Combo Boxes** checkbox is selected on the Options dialog, Standard Colors page, the last 16 color squares in the color selection palettes contain the project colors set on that dialog
- The **Font Color** field affects only the element name text; it does not override the feature text colors as defined on the Options dialog, Standard Colors page
- When it has been called directly on one or more selected elements (**Appearance | Set Font**), the Font dialog overrides the Default Appearance dialog
- When called from the Default Appearance dialog, the **Color** field on the Font dialog is overridden by the **Font Color** field on the Default Appearance dialog
- The **Font** and **Style** fields affect all text except for the Note compartment text
- The **Size** field affects all text

Learn More:

- [Standard Colors](#)424
- [Diagram Appearance](#)427
- [Format Toolbar](#)547

5.6.1.18 Get/Set Project Custom Colors

If more than one person is working on a project, you might want to share a set of custom colors with other team members, and possibly the whole project.

Access: [Settings | Colors | Set Project Custom Colors](#)
[Settings | Colors | Get Project Custom Colors](#)

Use to:

- Capture the project colors you have defined, in the project file, or
- Retrieve a set of colors someone else has defined, from the project file

How To:

To set your project's custom colors in the project file, follow the steps below:

Step	Action	See Also
1	Define the project colors on the Standard Colors page of the Options dialog	Standard Colors [424]
2	Select the Set Project Custom Colors option The following message displays: <i>Project Custom Colors export successful. 16 custom colors saved to the project file.</i>	
3	Click on the OK button	

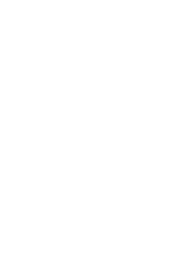
To get your project's custom colors from the project file, follow the steps below:

Step	Action	See Also
1	Select the Get Project Custom Colors option The following message displays: <i>Project Custom Colors import successful. 16 custom colors imported from the project file.</i>	
2	The colors are now made available in the color pallet of the Default Appearance dialog	Set an Element's Default Appearance [641]

5.6.1.19 Set Element Templates Package

Topics:

Topic	Detail	See also
Overview	<p>In building up a model, you might want to represent or emphasize certain characteristics of elements in the appearance of those elements, or select particular display options as standard</p> <p>For example, you could make new Interface elements a different default color to new Class elements, ensure all new Activity Partitions are vertical rather than horizontal, or set a specific group of display options for new diagrams; you could also define a set of characteristics to use for each development stage of a project</p> <p>To do all this, you create a diagram with all the characteristics you require, and store the diagram in an element Templates package; Enterprise Architect then checks this package:</p> <ul style="list-style-type: none"> • Whenever you start to create an element in a diagram or • Whenever you create a new diagram 	

Topic	Detail	See also
	<p>If it finds a template for that diagram type, Enterprise Architect applies the settings in that template to the new element or to the display options of the diagram; for example, you could save a diagram under the name <i>ClassTemplate</i>, to apply a set of display characteristics to all new Class elements and all new Class or Logical diagrams</p> <p>You should create the Templates package in an administrative View of the project file rather than in any work area, to prevent the package from being changed or lost in any project development work</p>	
<p>Modifying the appearance of elements</p> 	<p>There are two other ways in which you can modify the appearance of elements in diagrams:</p> <ul style="list-style-type: none"> You can define the default appearance of elements (and other structures) grouped in a diagram by using UML Profiles; these provide a means of extending the UML Language, which enables you to build UML models in particular domains <p>Profiles are based on additional stereotypes and Tagged Values that are applied to elements, attributes, methods, connectors and connector ends</p> <ul style="list-style-type: none"> You can modify the appearance of elements (and connectors) of a specific type using stereotypes, which take precedence over templates; if you drop an unsteriotyped element - a Class, for example - onto a diagram, Enterprise Architect searches the Templates package for a Class diagram that defines an unsteriotyped Class, and applies that definition to the new Class <p>If you drop a stereotyped Class onto a diagram, the stereotype defines the Class appearance so the template is not accessed; stereotypes are much more flexible for defining the appearance of an element under different scenarios</p>	<p>UML Profiles ^[1026]</p> <p>UML Stereotypes ^[1016]</p>

How To:

To set up the element Templates package, follow the steps below:

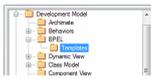
Image	Step	Action	See Also
	1	<p>Create a new package in the appropriate administration View</p> <p>You can give this package any name; <i>Templates</i> is an unambiguous option</p>	
	2	<p>Within the Templates package create new diagrams, one for each type of diagram to template</p> <p>Give them easily recognized names; for example <i>ClassTemplate</i> for the template for Class diagrams</p>	
	3	<p>Add new elements to the template diagrams from the Toolbox, and configure the size, appearance, notes, version and other properties</p>	
	4	<p>Select the Settings Project Template Package menu option to set the templates as the default element templates</p> <p>The Browse Project window displays</p>	

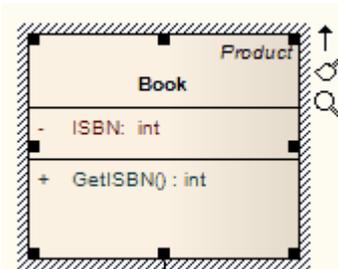
Image	Step	Action	See Also
	5	Locate and click on the Templates package, and click on the OK button to set the package as the default element template Now each new element or diagram you add to your project is created with the settings from the appropriate Template diagram	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Reference Data - Update** permission to set up or access the element Templates package
- If you decide not to use the default element template, set the default element template to **<none>** in the Browse Project window; the **<none>** package is at the bottom of the hierarchy shown in the Browse Project window

5.6.1.20 Highlight Context Element

You can show a hatched border around a selected element by selecting the **Always Highlight Context Element** checkbox on the Diagram Behavior page of the Options dialog. If you have selected this checkbox, the selected element displays similarly to the following example:



If you have not selected the **Always Highlight Context Element** checkbox, the selected element does not have a hatched border around it.

Topics:

Images	Topic	Detail	See also
	Multiple Selections	<p>Whether you have selected the Always Highlight Context Element checkbox or not, if you select multiple elements one of the elements you select always has a hatched border</p> <p>If you align the elements, this element is the one used to align the other elements against</p> <p>For example, if the elements in the diagram are aligned, the top element aligns to the bottom element (the element showing a hatched border)</p>	
	Change the Element to Align Against	To change which element has a hatched border in a selected group (and thus the element that is aligned with) click on the element that the other elements are to align with	

5.6.1.21 Make Linked Element a Local Copy

How To:

To convert a linked element to a local copy, follow the steps below:

Step	Action	See Also
1	Open the diagram with the linked element	
2	Select the linked element and right-click on it to display its context menu	
3	Select the Convert Linked Element to Local Copy menu option The element changes to a local copy and is placed in the appropriate package	

5.6.1.22 Copy Features Between Elements

Using drag and drop, you can copy attributes and/or operations from an element in the Project Browser on to another element in a diagram.

How To:

To copy an element feature, follow the steps below:

Images	Step	Action	See Also
	1	Open a diagram that contains the target element (in the example to the left, the AccountItem Class is the target and Customer element is the donor)	
	2	Click on the attribute or operation and drag it to the target element	
	3	Release the mouse button	

The image below shows *AccountItem* after the attribute *Account* has been dropped from the browser on to it.

To copy multiple element features, follow the steps below:

Images	Step	Action	See Also
	1	Open a diagram that contains the target element (in the example above, the AccountItem Class is the target and Customer element is the donor)	
	2	Hold down (Ctrl) (separate features) or (Shift) (select a range) and click on the attributes and/or operations to copy, then drag the selected features to the target element	
	3	Release the mouse button	

Learn More:

- [Attributes](#) ⁶⁹⁵
- [Operations](#) ⁷⁰⁷

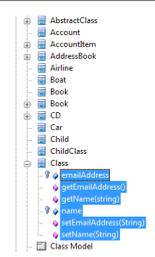
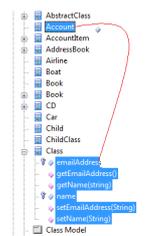
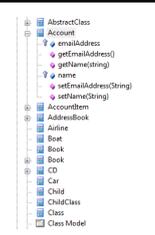
- [Move Features Between Elements](#)^[647]

5.6.1.23 Move Features Between Elements

Using drag and drop, you can move attributes and/or operations from an element in the Project Browser on to another element within the Project Browser.

How To:

To move element features, follow the steps below:

Images	Step	Action	See Also
	1	In the Project Browser, locate the attributes and/or operations to move from the target element and select them while holding down: <ul style="list-style-type: none"> • (Ctrl) (single item select) or • (Shift) (multiple item select) 	
	2	Holding down the mouse button, drag the attributes and/or operations to the target element A single feature symbol (and, under some operating systems, the feature name) displays during the move; however all of the selected features are moved	
	3	Release the mouse button The image to the left shows the final stage of the attribute and operations move between the Class element and the Account element	

Learn More:

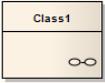
- [Attributes](#)^[695]
- [Operations](#)^[707]
- [Copy Features Between Elements](#)^[646]

5.6.1.24 Composite Elements

Enterprise Architect supports *Composite elements* for Classes, Objects and Use Cases. A Composite element is a pointer to a child diagram.

How To:

Images	Step	Action	See Also
	1	Create the element to set as a Composite element	

Images	Step	Action	See Also
	2	<p>Right-click on the element in the diagram and select the Advanced Make Composite context menu option</p> <p>The element displays (see Image on left)</p> <p>Note the small icon in the bottom right hand corner indicating that this is now a Composite element</p>	
	3	<p>Double-click on the Composite element to access the child diagram that it points to</p> <p>The Composite element and its child diagram are represented in the Project Browser as shown</p> <p>Note that <i>ClassX</i>, <i>ClassY</i> and <i>ClassZ</i> are elements in the child diagram</p>	

Topics:

Topic	Detail	See also
Alternative Notation	<p>Composite elements can show their contents instead of their usual notation</p> <p>To enable this:</p> <ol style="list-style-type: none"> 1. Right-click on the element to open the context menu 2. Select the Advanced Show Composite Diagram option 	
The Automation Interface	<p>Automation support is available for Composite elements</p> <p>Element has an <i>Elements</i> collection and a <i>Diagrams</i> collection</p>	Using the Automation Interface <small>(1822)</small>

Notes:

- If the **Make Composite** option is not listed in the context menu, the option is not available for the type of element you have selected

5.6.2 Element Context Menu

Right-click on a single element in a diagram to open the element context menu. If two or more elements are selected, a different, multiple selection context menu is displayed.

The element context menu is split into a number of sections and sub-menus:

- **Properties**
- **Add**
- **Find**
- **Transform (Ctrl+H)** - Transform the selected element from one domain to another
- **Embedded Elements**
- **Features**
- **Generate DDL** - Generate DDL for a table, procedure or view Class
- **Code Engineering**
- **Appearance**
- **UML Help** - display the Enterprise Architect Help topic for the UML element type
- **Delete (Ctrl+D)** - delete the element

Notes:

- Context menus vary between element types; for example, the **Code Engineering** options won't display for a Use Case element

Learn More:

- [Properties](#) ^[649]
- [Add Sub-menu](#) ^[651]
- [Find Sub-menu](#) ^[653]
- [Transform Elements](#) ^[1303]
- [Embedded Elements](#) ^[653]
- [Features Menu](#) ^[654]
- [Generate DDL](#) ^[1379]
- [Code Engineering Menu](#) ^[655]
- [Appearance Menu](#) ^[655]
- [Multiple Selection Context Menu](#) ^[658]

5.6.2.1 Properties Menu Section

The **Properties** section of the element context menu can contain the following options:

Action	Usage	Shortcut	See also
Properties	Open the Properties dialog for the selected element For State Lifeline and Value Lifeline elements, display the Configure Timeline dialog	Ctrl + Enter	Properties dialog ^[660]
Advanced	Open the Advanced sub-menu		Advanced Submenu ^[649]
Rule Composer	For a Rule Task element, invoke the Rule Composer tab in Business Rule Modeling		Compose Business Rules ^[1211]
Other Properties	For State Lifeline and Value Lifeline elements, display the Properties dialog for the selected element		Properties dialog ^[660]
Create (or Edit) Linked Document	(Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions) Create an RTF document linked to the element	Ctrl + Alt + D	Link Document to UML Element ^[730]
Delete Linked Document	Delete an existing linked document for the element		

5.6.2.1.1 Advanced Submenu

The **Advanced** submenu on an element context menu can contain the options listed in the table below.

Action	Usage	Shortcut	See also
Parent	Set the element parent	Ctrl + I	Set Element Parent ^[628]
Instance	Set the instance classifier for the element, on	Ctrl + L	Select <Item>

Action	Usage	Shortcut	See also
Classifier	the Select <Item> dialog		Dialog ^[690]
Classifier Properties	Open the Properties dialog for the <i>classifier</i> of the selected element	Ctrl + Alt + Enter	Properties Dialog ^[660]
Make Composite	Set the element as a Composite element		Composite Elements ^[647]
Change to State (Value) Lifeline	Switch one type of Lifeline element to the other		
Show Composite Diagram	Display a mini-picture of the contents of a composite element within that element		
Multiplicity	Define the multiplicity for the element, using the format defined on the Cardinality tab This is the number of instances of the element that can exist in a set; the value displays on the element in a diagram, in the <i>Name</i> compartment		Cardinality ^[776]
Show Element Type	(Ports and Parts only) Toggles between displaying and hiding the element type, as set on the Property page of the element's Properties dialog		Part ^[960] Port ^[962] The Property Page ^[963]
Edit Extension Points	For an extended Use Case, display the Use Case Extension Points dialog, which you use to insert the point at which the behavior should be inserted		Use Case Extension Points ^[937]
Association Class	Connect the Class to a new Association (if the element is a Class)		Connect New Class to Association ^[975]
Use Rectangle (Circle) Notation	Use rectangle notation for the element		Rectangle Notation ^[938]
Partition Activity	Define an Activity Partition		Activity Partition ^[876]
Set Run State	Add a new instance variable to the element using the Define Run State dialog		Define A Run-Time Variable ^[958]
Set Property Value	(Part elements) Set the property value for the Part, using the Set Property Values dialog		
Override Attribute Initializers	Pre-define initial values for attributes that can be used to override existing defaults	Ctrl + Shift + R	
Convert to Instance (Property)	Convert this classifier to an instance or a property, depending on the type of classifier selected (for example, SysML classifiers are always converted to properties)		
Convert Linked Element To Local Copy	Convert the occurrence of the element on this diagram from a link to the original element to a local copy of the element		
Make Sender/	Toggle the element from a sender to a receiver		

Action	Usage	Shortcut	See also
Receiver	and vice versa		
Accept Time Event	Change the notation for an Accept Event action to an Accept Time Event action		
Set Object State	Set the state of an object/instance based on the child states of its classifier	Ctrl + Shift + S	Object State ^[959]
Define Concurrent Substates	Define a set of substates that can be held simultaneously within that composite State		Regions ^[820]
Use State Label Notation	Display State Label Notation for a State object (the element name is displayed on a box on top of the element rather than inside it)		
Deep History	Change the type of a shallow History pseudo-state to a deep History Applies only when right-clicking on a History pseudo-state.		History ^[902]
Set Attached Links	Attach the selected Note element to a connector, or several connectors		Add A Note to A Connector ^[743]
Link to Diagram Note	Display the diagram notes as the Note element text The option simply deletes any current text and blocks the Note from being edited other than through the Notes field in the diagram Properties dialog		

Notes:

- Context menus vary between element types. Not all menu options shown here are present on all element context menus; for example, the Partition Activity option only displays for an Activity element
- If an element appearance is overridden by a Shape Script, several of the appearance options are disabled; for example, Use Rectangle (Circle) Notation

Learn More:

- [Shape Scripts](#) ^[1089]

5.6.2.2 Add Submenu

The **Add** submenu enables you to add supporting elements and diagrams to the selected element.

Menu Option	Action	Shortcut	See Also
Tagged Value	Add a Tagged Value		Quick Add - Tagged Value To Elements ^[763]
Related Elements	Open the Insert Related Elements dialog		Insert Related Elements ^[652]
Note	Create and attach a blank Note element to the current element		Create Notes and Text ^[639]

Menu Option	Action	Shortcut	See Also
Constraint	Create and attach a blank Constraint element to the current element		Note ^[913]
Activity	Add an Activity element as a child of the current Classifier element, with either an Activity diagram or an Interaction Overview diagram.		Activity ^[873] Object Classifiers ^[705]
Interaction	Add an Interaction element as a child of the current Classifier element, with either a Sequence diagram, a Communication diagram or a Timing diagram		Interaction ^[904]
State Machine	Add a State Machine element as a child of the current Classifier element, with a State Machine diagram		State Machine ^[925]
RuleFlow activity	For a Class element, create a Rule Flow Activity element with a child Rule Flow diagram, as a behavior for the Class		Create a Rule Flow Model ^[1206]
Add Diagram	Add a child diagram to the Classifier element, using the New Diagram dialog		Add New Diagrams ^[568]

Notes:

- Context menus vary between element types, so not all menu options shown here are present on all element context menus; for example, the options relating to Classifiers are not available for Object elements

5.6.2.2.1 Insert Related Elements

The Insert Related Elements dialog can be accessed from the **Add | Related Elements** option on most element context menus. This dialog enables you to insert connected elements from elsewhere in the model into the current diagram.

You can specify the following details:

Field	Usage	See also
Insert elements to: «x» levels	Select the level down to which to insert connected elements, between levels 1 and 5 You can select levels 4 or 5 to see how far the element/relationship hierarchy extends, but as this can produce a complicated and tangled diagram, it is better to use level 1 or 2 on selected elements in turn	
For Link Type	Select a type of connector to limit the inserted elements to those connected by that relationship type	
With Link Direction	Select whether the connectors are to be a single direction or bi-directional	
Limit to Element Type	Select a type of element to limit the inserted elements to those of that element type	
Layout Diagram When Complete	Select whether Enterprise Architect should layout the diagram after the elements have been inserted; the layout applied is the Digraph	Digraph Layout ^[612]

Field	Usage	See also
	layout If no elements have been added, this option has no effect; elements have to be added for Enterprise Architect to adjust the layout	
Limit to this Namespace	Select a specific namespace from which the inserted elements are to come	

5.6.2.3 Find Submenu

The **Find** submenu on the element context menu can contain the following options:

Action	Usage	Shortcut	See also
In Project Browser	Highlight the currently selected element in the Project Browser	Alt + G	
Locate Classifier In Project Browser	Highlight the classifier for the currently-selected object, in the Project Browser	Ctrl + Alt + G	
Locate Operation in Project Browser	Highlight the call operation for the currently-selected Activity, in the Project Browser	Ctrl + Alt + G	Class Operations in Activity Diagrams [872]
In Diagrams	Open the Element Usage dialog	Ctrl + U	Show Element Use [630]
Custom References	Set up cross references	Ctrl + J	Set Up Cross References [630]
Add to Favorites	Add the element to the Favorites folder in the Resources window		Favorites [790]

5.6.2.4 Embedded Elements Submenu

The **Embedded Elements** submenu on the element context menu can contain the following options:

Menu Option	Action	Shortcut	See also
Add Port	Add an embedded Port to the element		
Add Required Interface	Add an embedded Required Interface to the element		
Add Provided Interface	Add an embedded Provided Interface to the element		
Add Action Pin	Add an embedded Action Pin to the element		
Add Expansion Node	Add an embedded Expansion Node to the element		
Add Object Node	Add an embedded Object Node to the element		
Add Activity Parameter	Add an embedded Activity Parameter to the element		

Add Entry Point	Add an embedded Entry Point to the element		
Add Exit Point	Add an embedded Exit Point to the element		
Embedded Elements	Open the Embedded Elements window		Embedded Elements Window ^[654]
Show Realized Interfaces	Display each interface directly realized by a Class		Show Realized interfaces of Class ^[597]
Show Dependent Interfaces	Display each dependency relationship for that model element as a lollipop style node attached to its left-hand side		

Notes:

- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, of the **Add** options, only **Add Port** displays for a Class element

5.6.2.4.1 Embedded Elements Window

The **Embedded Elements** dialog enables you to embed particular elements into other elements. For example, a Port can be embedded into a Class. The **Embedded Elements** option is available on the context menu of some elements.

Topics:

Image	Topic	Detail	See also
	Usage	<p>In the Embedded Elements dialog, click on the New button to create a new embedded element</p> <p>Enter details such as type, name and stereotype, and click on the OK button</p> <p>The embedded element now shows on the primary element as shown</p> <p>You can add as many embedded elements as necessary; modify or delete embedded elements using the Embedded Elements dialog</p> <p>To incorporate inherited or owner properties, select the Show Owned/Inherited checkbox</p> <p>The name of the embedded element is a label, which you can edit using the Labels context menu</p>	Label Menu Section ^[597]

5.6.2.5 Features Menu Section

The **Features** section of the element context menu can contain the following options:

Action	Usage	Shortcut	See also
Attributes	Open the Attributes ^[695] dialog		Attributes ^[695]

Action	Usage	Shortcut	See also
Operations	Open the Operations [707] dialog		Operations Dialog - General [708]
Feature Visibility	Open the Feature Visibility [489] dialog	Ctrl + Shift + Y	Feature Visibility [585]

Notes:

- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, the **Attributes** and **Operations** options won't display for an Action element

5.6.2.6 Code Engineering Menu Section

The **Code Engineering** submenu on the element context menu can contain the following options:

Action	Usage	Shortcut	See also
Generate Code	Generate source code [1497] for the selected element (forward engineer)	F11	Generate a Single Class [1497]
Synchronize With Code	Reverse engineer source code [1508] for the selected element	F7	Importing Source Code [1508]
View Source Code	Open the source editor [1408] if a file exists for that selected element	F12	The Source Code Viewer [1408]
Create Workbench Instance	Create a workbench instance for the Debug Workbench (if a debug command [1665] has been configured for the parent package)	Ctrl + Shift + J	Create Workbench Variables [1665]

Notes:

- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, Code Engineering options won't appear for a Use Case element

5.6.2.7 Appearance Menu Section

The **Appearance** section of the element context menu can contain the following options:

Reference:

Menu Option	Action	Shortcut	See also
Lock Element	<p>Lock the element so it can't be edited; to unlock the element, select Lock Element again</p> <p>This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions when security is enabled; in that situation, use the Lock option for model elements</p>		Lock model elements [210]

Menu Option	Action	Shortcut	See also
Selectable	<p>Toggle whether the element is selectable or not</p> <ul style="list-style-type: none"> If an element is selectable, you can move it around the diagram and perform right-click context menu operations If an element is unselectable, you cannot move it around the diagram and the only right-click operation available is to make the element selectable <p>This option has no effect on double-click operations on the element, such as displaying child diagrams or Properties dialogs</p> <p>An element on a locked diagram is also unselectable - if you click on it, the element outline displays in red</p> <p>In a diagram, you can filter the display to show selectable or non-selectable elements only</p>		Diagram Filters ^[489]
Dockable	<p>Align and join two elements either vertically or horizontally, on the current diagram only</p> <p>Both elements must have the Dockable option selected, and must have the joining edges parallel</p> <p>As the distance between the elements narrows, the moving element snaps to the edge of the other element</p> <p>For Activity Partitions, the option is selected by default.</p> <p>Deselecting the Dockable option does not separate the elements; if necessary, you can simply move the elements apart again</p>		Partition ^[914]
Appearance	Display the Appearance submenu; see the table below		
Z-Order	Set the Z-Order of the element		Z-Order Elements ^[584]

Appearance Sub-Menu

Action	Usage	Shortcut	See also
Default Appearance	<p>Override the <i>global</i> default appearance of all elements (which you set on the Options dialog, Standard Colors page and Diagram Appearance page) with a different default for just the selected element on all diagrams in which it is found</p> <p>To change the appearance of the selected element on the <i>current diagram only</i>, use the Format toolbar</p>	F4	Standard Colors ^[424] Appearance ^[427] Set an Element's Default Appearance ^[64] Format toolbar ^[54]
Apply Image From	Paste the image held on the clipboard onto the		

Action	Usage	Shortcut	See also
Clipboard	selected element		
Select Alternate Image	Select an alternative image using the Image Manager	Ctrl + Shift + W	Using the Image Manager [593]
Hide/Show Name Under Image	Hide or redisplay the name label under an element with an alternative image		
Set Font	Change the font type, size, color and effects for the text in an element		Set Element Font [657]
Show Labels	Reveal any hidden labels on the element		
Copy Appearance to Painter	Copy the default element appearance (set using the Default Appearance option, above) to the painter You then paste the default appearance using the Paste Appearance option on the Diagram toolbar		Diagram toolbar [117]
Copy Image to Clipboard	Copy the element image to the clipboard		

Notes:

- You can also change the appearance (and other aspects) of several selected elements at once
- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, the **Alternate Image** option won't display for a Lifeline element

Learn More:

- [Element Multiple Selection Menu](#) [658]

5.6.2.7.1 Set Element Font

Topics:

Topic	Detail	See also
Usage	<p>You can change the appearance of the text within an element, for one or more selected elements</p> <ul style="list-style-type: none"> • Selecting the Appearance Set Font context menu option • Selecting the Font icon on the Format toolbar, or • Selecting the Font Style option on the element Default Appearance dialog (for the selected element, overridden by the two options above). <p>The Font dialog displays</p> <p>Select the font, style, size, color and effects, and (if necessary) the script type. Click on the OK button to save your changes</p>	Format Toolbar [547] Default Appearance [647]

5.6.2.8 Element Multiple Selection Menu

You can perform operations on two or more elements on a diagram at once. To select the required elements, either click and drag the cursor over the group to highlight them, or press (**Shift**) and click on each element.

Topics:

Topic	Detail	See also
Usage	<p>Right-click on an element to display a context menu that enables you to:</p> <ul style="list-style-type: none"> Align elements (by left edge, right edge, top, bottom, center in a column or center in a row) Space elements evenly (across or down) Standardize the dimensions of the selected elements Specify the visibility of features for all selected elements Add the same Tagged Value to all selected elements Automatically resize elements to match (element content permitting) Turn the Dockable option on or off for all selected elements on a diagram Set the default appearance and font for multiple elements at once Make the selected elements on the diagram non-selectable; to make them selectable again, right-click on the diagram and select the Make All Elements Selectable context menu option Generate code for all selected elements at once, or synchronize the code against the selected elements Transform the selected elements Copy all selected elements to the clipboard Automatically adjust the layout of the selected elements on the diagram Delete all selected elements 	<p>Feature Visibility ^[585]</p> <p>Assign a Tagged Value to an Item ^[764]</p> <p>Appearance Menu Section ^[656]</p> <p>Set an Element's Default Appearance ^[641]</p> <p>Set Element Font ^[657]</p> <p>Transform Elements ^[1305]</p>

Notes:

- Where elements are made the same, they are matched to the element you right-clicked on
- It is much faster to assign an appearance or characteristic to a group of elements than to one element at a time

5.6.3 Visual Representation

Each UML element has a default representation. However, you can:

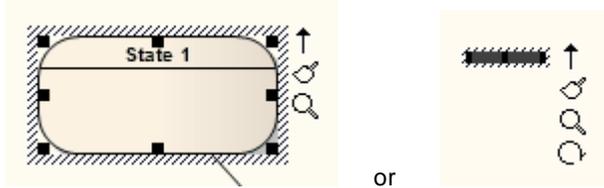
- Change the appearance, features, position and - for some elements - orientation through a number of toolbars
- Display or hide various types of information held in the element, in compartments

Learn more:

- [Element Icons](#) ^[659]
- [Compartments](#) ^[659]

5.6.3.1 Element Icons

When you add an element to a diagram, or select an existing element, a number of small icons display off the right hand side of the element, underneath the Quicklinker arrow. For example:



These icons display small versions of the diagram toolbars or perform specific actions, to enable you to quickly edit the element you have highlighted.

Reference:

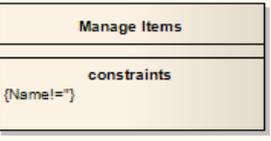
Icon	Description	See Also
	Rotates a Fork/Join element from vertical to horizontal and vice-versa	
	Displays the Format toolbar, for changing element appearance	Format Toolbar [54↑]
	Displays the Current Element toolbar, to edit the element's properties and features	Current Element Toolbar [112↑]
	When multiple elements are selected, displays the Diagram Toolbar for changing or aligning the elements together	Diagram Toolbar [11↑]
	Toggles an Activity Partition between vertical and horizontal	Activity Partition [876]

5.6.3.2 Compartments

In addition to the attributes and operations compartments shown in a Class element, Enterprise Architect enables you to display or hide other compartments.

Topics:

Images	Topic	Detail	See also
<ul style="list-style-type: none"> Or, fully qualified, expanded format: 	Tags Compartment	<p>The <i>tags</i> compartment lists all Tagged Values for an element as entered in the Tagged Values window</p> <p>The fully-qualified option operates only on those Tagged Values that were created in Enterprise Architect release 7.1 or later; it does not expand Tagged Values created in earlier releases</p>	Tagged Values [762]

Images	Topic	Detail	See also
	Responsibilities Compartment	The <i>responsibilities</i> compartment shows a list of responsibilities as entered on the Requirements page of the element Properties dialog	Requirements ^[664]
	Constraints Compartment	The <i>constraints</i> compartment shows a list of element constraints as entered in the Constraints page of the element Properties dialog	Constraints ^[666]
	Testing Compartment	The <i>testing</i> compartment lists all of the tests associated with an element as listed in the Testing window	The Testing Workspace ^[1695]
	Maintenance Compartment	The <i>maintenance</i> compartment lists all of the defects, changes, issues and tasks associated with an element, as listed in the Maintenance window	The Maintenance Workspace ^[1713]

Notes:

- To set the visibility of the various compartments, see *Feature Visibility*

Learn More:

- [Feature Visibility](#) ^[585]

5.6.4 Element Property Displays

To define, assign and review the properties and features of elements, you can use the following:

Topic	Link
Properties Dialog	Properties Dialog ^[660]
Properties Docked Window	Properties Window ^[689]
Element Browser	The Element Browser ^[687]
Scenarios & Requirements Window	The Scenarios & Requirements Window ^[689]
Select <Item> Dialog	Select <Item> Dialog ^[690]

5.6.4.1 Properties Dialog

This topic area covers element properties and their settings, responsibilities, constraints, connectors, scenarios, Tagged Values, associated files, object files and classifiers, and boundary element settings.

Topics:

Topic	Detail	See also
Displaying the Properties dialog	<p>To display the element Properties dialog, use any of the following methods:</p> <ul style="list-style-type: none"> • Select an element in the Diagram View and select the Element Properties menu option • Right-click on an element in the Diagram View, and select the Properties context menu option • Select an element in the Diagram View, and press (Alt+Enter) • Double-click on an element in the Diagram View • Right-click on an element in the Project Browser, and select the Properties context menu option <p>To suppress display of the Properties dialogue when placing a new element, uncheck the Edit Object on New option on the Objects page of the Options dialog</p>	Objects ^[432]
Properties dialog pages	<p>The Class Properties dialog consists of the following pages:</p> <ul style="list-style-type: none"> • General • Details • Advanced Properties • Templates • Tagged Values • Requirements • Constraints • Scenarios • Files • Links <p>Properties also include Object files and Classifiers, and the Boundary element appearance</p>	General Settings ^[662] Details ^[663] Advanced Properties ^[664] Templates ^[664] Tagged Values ^[762] Requirements ^[664] Constraints ^[666] Scenarios ^[667] Associated Files ^[687] Links ^[666] Object Classifiers ^[705] Boundary Element Settings ^[932]

Notes:

- There are several variations of the Properties dialog:
 - The dialog for a Table or Stored Procedure element has slight differences on the General page, and a Table (Stored Procedure) Details page instead of a Details page
 - The dialog for a Class element of a stereotype other than Table is as shown in General Settings
 - The dialog for an element of any other type does not have a Details page
 - Port and Part elements have a Property page
 - Activity elements have a Behavior page, and Action and Invocation elements (depending on their type) have other pages such as Effect, Trigger and Call pages
 - Action Pins have a Pin page
- In all cases, the Properties dialog is an expandable window, which you can stretch to enable longer entry and clearer inspection of the text field values
- The Tagged Values page of the element Properties dialog simply provides the Tagged Values window within the frame of the Properties dialog.

Learn More:

- [Working With Table Properties](#) ^[1348]

- [Port](#)^[962]
- [Part](#)^[960]
- [The Property Page](#)^[963]
- [Interactions and Activities](#)^[715]
- [Action](#)^[865]
- [Trigger](#)^[934]
- [Behaviour Calls](#)^[715]
- [Action Pin](#)^[869]

5.6.4.1.1 General Settings

The General page of the element Properties dialog contains the following fields:

Field	Usage	See also
Name	Change the element's name	
Stereotype	(Optional) Type the name of a stereotype for the element, or click on the drop-down arrow and select one	
Role	Type the role played by the object represented by the element	
Alias	Type an alias (alternative display name) for the object	
Keywords	Enter free-text items such as keywords or context information; this can be filtered in Use Case Metrics and Search dialogs	
Author	Enter or select the name of the original author	
Language	Select the programming language for the object	
Status	Indicate the current status of the element (such as Approved or Proposed)	
Complexity	Indicate the complexity of the element (used for project estimation), such as Easy, Medium or Hard	
Version	Enter the version of the current element	
Phase	Indicate the phase this element is to be implemented in (for example, 1, 1.1, 2.0 ...)	
Notes	Enter any notes text associated with the element, as described for the Notes window You can format the notes text using the Notes toolbar at the top of the field	Notes ^[769] Notes Toolbar ^[770]

Further facilities are made available by clicking on the **Advanced** page.

Learn More:

- [Advanced Settings](#)^[664]

5.6.4.1.2 Details

The **Details** page of the element **Properties** dialog enables you to define the structural and processing details for the selected Class element.

Reference:

Field/Button	Usage	See also
Scope	Click on the drop-down arrow and select the element's scope (public, private, protected, package)	
Persistence	Click on the drop-down arrow and select the appropriate value (blank, persistent or transient)	
Cardinality	The number of instances of the element that can exist - the value displays on the element in a diagram, in the <i>Name</i> compartment Use the format defined in the Cardinality tab	Cardinality [776]
Abstract	Select if the element is abstract	
Is Root	Select if the element is a root element and cannot be descended from another	
Is Leaf	Select if the element is final and cannot be a parent for other elements	
Is Specification	Select if the element is a specification	
Is Active	Select if the element is active; for example, an Active Class	Active Classes [943]
Attributes	Define attributes for the Class The Attributes Properties dialog displays	Attributes [695]
Operations	Define operations for the Class The Operations Properties dialog displays	Operations Dialog - General [708]
Concurrency	Specify how concurrent activities should be processed	
Collection Classes	Define Collection Classes (for generating code from Association connectors) that apply to this Class The Collection Classes for Association Roles dialog displays	Set Collection Classes [1526]
Template Parameters	List the <i>Parameterized Class</i> template parameters Click on the Edit button to edit a selected parameter, or click on the Add button to add a new parameter	Parameterized Classes [943]
Binding(s)	List the binding expressions on a <i>binding</i> Class Click on the Edit button to edit a selected binding expression, or click on the Add button to add a new binding expression	Template Binding [1010]

Notes:

- When launched from MDG Integration, the **Attributes** and **Operations** buttons are not available

5.6.4.1.3 Advanced Properties

The Advanced page displays the advanced properties of the element (where they exist) and enables you to reset the values of these properties.

How to:

To set a value for an advanced (or custom) property, follow the steps below:

Step	Action	See also
1	Click on the data field to the right of the property name	
2	Depending on the property, either: <ul style="list-style-type: none"> • Type the value in free text • Click on the drop-down arrow and select the value from the list • Click on the (...) Browse button and search for the required value 	
3	Click on the Apply button	

5.6.4.1.4 Templates

Use the Templates page to define

- Parameterized Class template parameters
- Binding expressions on a binding Class

In the Template Parameter(s) panel, click on the **Add** button to add a new parameter, or click on the **Edit** button to edit a selected existing parameter; in either case, the Template Parameter dialog displays.

In the Binding(s) panel, click on the **Add** button to add a new binding expression or click on the **Edit** button to edit a selected binding expression. When you click on the **Add** button, a short context menu displays from which you select the type of relationship to generate, and then you define the binding expressions.

If either a template parameter or a binding expression is no longer required, click on it and then click on the corresponding **Delete** button.

Learn More:

- [Parameterized Classes \(Templates\)](#)^[943]
- [Template Binding](#)^[1010]

5.6.4.1.5 Requirements

You use the Requirements page of the element Properties dialog to create requirements that this element is designed to meet.

Requirements are of two types:

- Internal requirements (responsibilities) and
- External requirements (system requirements, elements connected to the element by a Realize connector)

Enterprise Architect shows both types, but you can only edit the internal type from the Requirements page.

You can show the requirements for an element on the diagram directly, using the Feature Visibility function; it is also possible to show inherited requirements in this way.

Field/Button	Usage	See Also
Requirement	Enter the name and high level detail of the requirement	
Type	Specify the type; for example, Functional or Non-functional: <ul style="list-style-type: none"> Functional requirements are things that the system must do, such as identify franked, unfranked and total credit for a dividend Non-functional requirements are things that the system must be, such as reliable or cost effective 	
Last update	Specify the date of the last requirement update	
Status	Specify the current status of the requirement	
Difficulty	Identify the difficulty of implementing the current requirement	
Priority	Specify how urgent the requirement is	
Stability	Specify the estimated stability of the requirement This is an indication of the probability of the requirement - or understanding of the requirement - changing; high stability indicates a low probability of the requirement changing	
Notes	Record details of the requirement You can format the notes text using the Notes toolbar at the top of the field	Notes Toolbar ^[770]
Move External	Make an internal responsibility into an external requirement	Make Internal Requirement External ^[1167]
New	Create a new requirement	
Save	Save changes to requirements	
Delete	Delete a selected requirement	
Defined	List the defined requirements associated with this element	

Learn More:

- [Internal Requirements](#)^[1167]
- [External Requirements](#)^[665]
- [Feature Visibility](#)^[585]

5.6.4.1.5.1 External Requirements

External requirements are those Requirement elements that have been connected to the current element using a *Realization* connector. By creating the connector from the element to the requirement, you create an expectation that the element must implement the requirement as part of the system solution.

In Enterprise Architect, linked requirements are shown in the Requirements page of the element Properties dialog, but they are marked external and cannot be directly edited (on selection, the page fields are grayed out).

Double-click an external requirement in the list to activate the Properties dialog for the associated requirement, where you can view and modify the requirement details and check the requirement hierarchy details.

Learn More:

- [Create Requirements](#) ^[1156]
- [Requirement Properties](#) ^[1157]
- [Make Internal Requirement External](#) ^[1167]

5.6.4.1.6 Constraints

Elements can have associated constraints placed on them. Constraints are used in conjunction with responsibilities to define the conditions and rules under which an element operates and exists.

Typical constraints are pre- and post- conditions, which indicate things that must be true before the element is created or accessed and things that must be true after the element is destroyed or its action complete.

Use the Feature Visibility function to show constraints for an element on the diagram directly; it is also possible to show inherited constraints in this way.

How To:

To add constraints to a model element, follow the steps below:

Step	Action	See Also
1	Open the element Properties dialog	
2	Select the Constraints page	
3	In the Constraint field, type the name of the constraint	
4	In the Type and Status fields, click on the drop-down arrow and select the constraint type (Pre-condition , Post-condition or Invariant) and status	
5	In the larger text field, type any additional notes required	
6	Click on the Save button	

Learn More:

- [Feature Visibility](#) ^[585]
- [Requirements](#) ^[664]

5.6.4.1.7 Links

The Links page of the element Properties dialog displays a list of all relationships active for the current element.

Topics:

Topic	Detail	See also
Relationships Panel	The Relationships panel lists the relationships this element has, including: <ul style="list-style-type: none"> • Elements this element is related to • Element Stereotype (if any) of the element • Type of the related element 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • Connection or relationship type • Stereotype (if any) of the relationship 	
Links Page	<p>From the Links page you can perform operations on a relationship, by right-clicking on the relationship to display the context menu</p> <p>To:</p> <ul style="list-style-type: none"> • Hide the relationship on the diagram, click on the Hide Relation menu option; the option then changes to Show Relation, which you select to redisplay the relationship on the diagram • Display the relationship Properties dialog, click on the Relationship Properties menu option • Highlight the related element in the Project Browser, click on the Locate Related Object menu option • Delete the relationship from the model and all diagrams, click on the Delete Relationship menu option; the system prompts you to confirm the deletion 	Connector Properties <small>[756]</small>

5.6.4.1.8 Scenarios

A scenario is a real-world sequence of operations that describes how an element works in real-time.

Topics:

Topic	Detail	See also
Where you can generate Scenarios	<p>You can apply scenarios to any element - generally Use Cases - to describe functional behavior, business work flows and end-to-end business processes</p> <p>You create scenarios through the Scenario page of the element Properties dialog; this page has two internal tabs, as described below.</p> <ul style="list-style-type: none"> • The Description tab enables you to create scenarios and provide a simple text description of each scenario, or of the structure of each scenario • The Structured Specification tab (the default) enables you to create scenarios or select those you have created elsewhere and, for each scenario: <ul style="list-style-type: none"> • Create a series of steps for each part of the scenario • Structure the scenario to show how the basic path diverges into the alternate paths and exception paths • Generate a number of types of diagram from the structure • Generate a structured scenario from an Activity diagram • Generate a structured scenario from text on the clipboard; this option has a variation in the Description tab that enables you to translate scenario descriptions created prior to release 8.0 of Enterprise Architect, into structured scenarios in the latest release 	Structured Specification Tab <small>[669]</small> Generate Diagrams <small>[676]</small> Generate Scenario from Activity diagram <small>[684]</small> Structured Specification Item Context Menu <small>[672]</small> Scenarios <small>[668]</small>
Description Tab	On the Scenarios page, click on the Description tab; the	

Topic	Detail	See also
	<p>Scenario (name) field and the Type field both default to Basic Path, to enable you to define the basic path first; you can overtype the scenario name with more appropriate text if required</p> <p>As you go on to create other scenarios, you set the type to Alternate or Exception as appropriate</p> <p>Complete the fields as described in the table below</p>	

Reference:

Field	Usage	See also
Scenario	Type in the name of the scenario (or, for existing scenarios, click on the drop-down arrow and select one from the list)	
Type	Specify the type of scenario; the options are: <ul style="list-style-type: none"> • Basic Path - the direct set of steps for the scenario • Alternate - an alternative set of steps, in parallel with part of the basic path • Exception - the path the scenario follows if a step of the basic path does not produce an appropriate result 	
Description	<p>Record a textual description of how the user uses the current element</p> <p>As for the Notes window, you can format the notes text using the Notes toolbar at the top of the field</p> <p>As well as the Notes window facilities, you can also generate a structured specification from the text in this field; highlight the text, right-click on it and select the Create Structure from Notes context menu option</p> <p>The text is copied to the Structured Specification tab for the current scenario, either as a new specification or as the continuation of an existing specification, with a new step created at each carriage return</p> <p>Subsequent changes to the text on the Structured Specification tab are not reflected on the Description tab</p>	Note Tab ^[769] Notes Toolbar ^[770] Structured Specification Tab ^[669]
New	Clear the data fields so that you can enter data in them to create a new scenario	
Save	Save a new scenario, or changes to an existing scenario	
Delete	Delete a scenario selected from the Scenarios panel, below	
Scenarios	<p>Display a list of defined scenarios for this element</p> <p>You can change the order in which the scenarios are listed, using the  and  buttons</p>	

Notes:

- The Scenario page does not prevent you from creating more than one basic path, but it would be

unusual to define more than one

- All the functions available on the Scenario page are also available through the Scenarios & Requirements window/view; use the **Browse Element** icon in the window toolbar to list and select the scenarios for the element

Learn More:

- [Scenarios & Requirements](#) ^[689]

5.6.4.1.8.1 Structured Specification Tab

The Structured Specification tab offers a wide range of facilities for generating and modifying scenario specifications, enabling you to define the structure, actions and interactions of the scenarios defined for an element such as a Use Case. These scenarios can be the main (basic) path, alternate paths or exception paths.

When you open the Structured Specification tab it defaults to the basic path so that if it does not already exist, you can create it. You can create alternate paths and exception paths as part of the process of adding them to steps of the basic path.

You can also create all three types of scenario paths on the Description tab, or in the Scenarios & Requirements window.

Learn More:

- [Scenarios](#) ^[667]
- [Scenarios & Requirements](#) ^[689]
- [Structured Specification Toolbar](#) ^[671]
- [Item context menu](#) ^[672]
- [Selected Text context menu](#) ^[674]
- [Entry Points context menu](#) ^[675]
- [Floating Toolbar](#) ^[676]

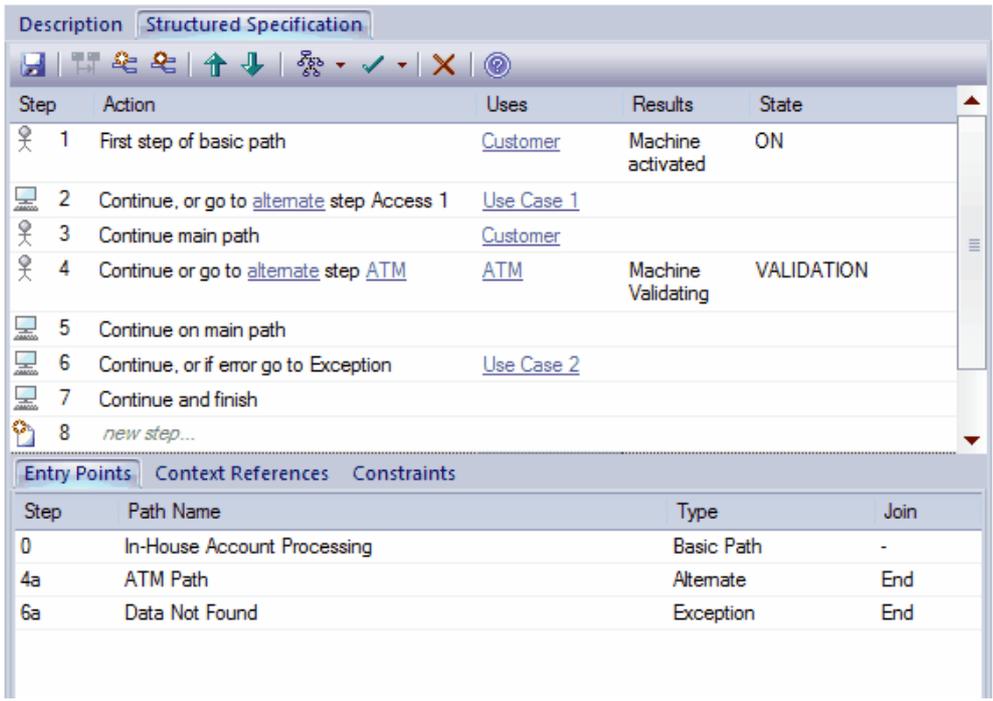
You can create a specification for a scenario in one of several ways:

- Enter the specification yourself, as described below
- Generate a specification from an Activity diagram created under a Use Case element
- Generate a specification from the notes text of the scenario in the Description tab
- Generate a specification from text held on the clipboard

How To:

To enter the specification yourself, starting with the basic path, follow the steps below:

Step	Action	See Also
1	In the Scenario field, click on the drop-down arrow and select the <i>Basic Path</i> scenario	
2	In the <i>new step</i> field in the Action column, type the text of the first step or action	
3	Tab to the Uses column and, if necessary, type the name of each element used in this step; any elements that are listed in the Context References tab are highlighted in blue and underlined You can also manage the list of elements as context references, and add to the list directly; see the Notes below	

Step	Action	See Also																																																													
4	Tab to the Results column and, if necessary, type the outcome of completing this step																																																														
5	Tab to the State column and, if necessary, type the name of the state into which the step moves the action																																																														
6	<p>When you move out of the Action column, the next <i>new step</i> field displays underneath</p> <p>Repeat steps 2 to 5 as many times as is necessary.</p> <p>The Structured Specification tab should now resemble the following illustration:</p>  <p>The screenshot shows the 'Structured Specification' tab with a table of steps and an 'Entry Points' sub-tab. The main table has columns: Step, Action, Uses, Results, and State. The 'Entry Points' sub-tab has columns: Step, Path Name, Type, and Join.</p> <table border="1" data-bbox="323 734 1316 1097"> <thead> <tr> <th>Step</th> <th>Action</th> <th>Uses</th> <th>Results</th> <th>State</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>First step of basic path</td> <td>Customer</td> <td>Machine activated</td> <td>ON</td> </tr> <tr> <td>2</td> <td>Continue, or go to <u>alternate</u> step Access 1</td> <td>Use Case 1</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>Continue main path</td> <td>Customer</td> <td></td> <td></td> </tr> <tr> <td>4</td> <td>Continue or go to <u>alternate</u> step ATM</td> <td>ATM</td> <td>Machine Validating</td> <td>VALIDATION</td> </tr> <tr> <td>5</td> <td>Continue on main path</td> <td></td> <td></td> <td></td> </tr> <tr> <td>6</td> <td>Continue, or if error go to Exception</td> <td>Use Case 2</td> <td></td> <td></td> </tr> <tr> <td>7</td> <td>Continue and finish</td> <td></td> <td></td> <td></td> </tr> <tr> <td>8</td> <td><i>new step...</i></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <table border="1" data-bbox="323 1108 1316 1288"> <thead> <tr> <th>Step</th> <th>Path Name</th> <th>Type</th> <th>Join</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>In-House Account Processing</td> <td>Basic Path</td> <td>-</td> </tr> <tr> <td>4a</td> <td>ATM Path</td> <td>Alternate</td> <td>End</td> </tr> <tr> <td>6a</td> <td>Data Not Found</td> <td>Exception</td> <td>End</td> </tr> </tbody> </table>	Step	Action	Uses	Results	State	1	First step of basic path	Customer	Machine activated	ON	2	Continue, or go to <u>alternate</u> step Access 1	Use Case 1			3	Continue main path	Customer			4	Continue or go to <u>alternate</u> step ATM	ATM	Machine Validating	VALIDATION	5	Continue on main path				6	Continue, or if error go to Exception	Use Case 2			7	Continue and finish				8	<i>new step...</i>				Step	Path Name	Type	Join	0	In-House Account Processing	Basic Path	-	4a	ATM Path	Alternate	End	6a	Data Not Found	Exception	End	
Step	Action	Uses	Results	State																																																											
1	First step of basic path	Customer	Machine activated	ON																																																											
2	Continue, or go to <u>alternate</u> step Access 1	Use Case 1																																																													
3	Continue main path	Customer																																																													
4	Continue or go to <u>alternate</u> step ATM	ATM	Machine Validating	VALIDATION																																																											
5	Continue on main path																																																														
6	Continue, or if error go to Exception	Use Case 2																																																													
7	Continue and finish																																																														
8	<i>new step...</i>																																																														
Step	Path Name	Type	Join																																																												
0	In-House Account Processing	Basic Path	-																																																												
4a	ATM Path	Alternate	End																																																												
6a	Data Not Found	Exception	End																																																												

Notes:

- By default, the steps begin with a user step (indicated by an actor icon) and alternate between user and system (indicated by a screen icon) steps. However, you can control the responsible entity assigned to a step by typing a keyword within the first 15 characters of the text in the Action column; either 'User' or 'Actor' for a user step, or 'System' for a system step. Once a step has been saved, you can change the responsible entity by either double-clicking on the icon or right-clicking on the step, and selecting the **Set Step as 'System'** or **Set Step as 'User'** context menu option as required
- An entry for the basic path displays in the Entry Points tab, as **Step 0** with no value in the **Join** column (the basic path does not rejoin itself)
- The values in the **Uses**, **Results** and **State** columns, whilst optional, are significant if you want to generate a diagram from the specification; if you type the name of an element linked to the current element (and listed in the Context References tab), the element name is highlighted and underlined
- In the **Uses** column, in edit mode, the **Manage Uses Context List** context menu option enables you to enable and disable **Uses** cells as linked element lists as well as text fields, across the whole model; when the feature is enabled, the column shows both context-linked elements and manually-entered text, and when disabled only text entries are listed (the context-linked entries are hidden and deactivated, but not deleted - they are re-activated when the feature is enabled)
- As you develop the scenario, you can move steps to different positions in the scenario, and insert new step lines within the body of the scenario
- You can also create constraints on the element that have an impact on the scenarios

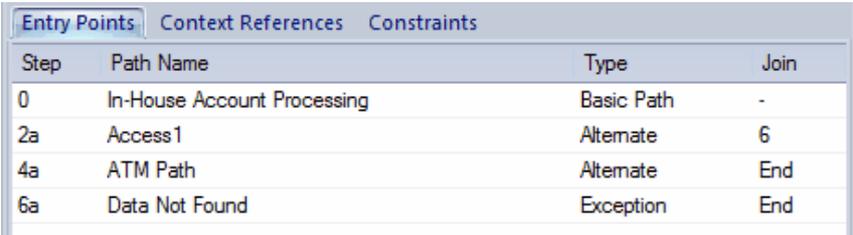
- Repeat the **Scenario Steps** procedure for each scenario you have created; you can now adapt, enhance and interrelate the scenario specifications using the facilities of the Structured Specification tab:
 - Structured Specification Toolbar
 - Item context menu
 - Selected Text context menu
 - Entry Points context menu
 - Floating Toolbar

Learn More:

- [Generate a Scenario from an Activity diagram](#) ^[684]
- [Scenarios](#) ^[667]
- [Structured Specification Item Context Menu](#) ^[672]
- [Generate Diagrams](#) ^[676]
- [Context References Tab](#) ^[686]
- [Structured Specification Toolbar](#) ^[671]
- [Scenario Constraints Tab](#) ^[686]
- [Structured Specification Entry Points Context Menu](#) ^[675]
- [Structured Specification Floating Toolbar](#) ^[676]
- [ScenarioStep Class](#) ^[1894]

The icons on the **Structured Specification** toolbar offer the following facilities:

Icon	Action	See also
	Save changes to the scenario specification.	
	Return to the basic path specification (if another specification is currently displayed).	
	<p>(Only enabled when the basic path is displayed - you cannot add an alternate path to another alternate path or an exception path.)</p> <p>Create a branch from the selected step to an alternate path scenario - select the path from the displayed list. If the appropriate scenario does not yet exist, double-click on the <i>new path</i> line and type the scenario name, then click off the line and back on to it. Click on the OK button.</p> <ul style="list-style-type: none"> • An entry for this alternate path displays in the Entry Points tab, as Step a of the basic path step it branches from; in the Join column, click on the drop-down arrow and select the number of the step at which action flows back to the basic path, or select End if the path terminates separately from the basic path • You cannot add more than one branch - whether alternate or exception - to a step • You cannot join the end of the branch - whether alternate or exception - to more than one step of the basic path. If you change the join point of the branch, any duplicates of that branch are updated to remain in synchrony with the join point 	

Icon	Action	See also																				
	 <table border="1"> <thead> <tr> <th>Step</th> <th>Path Name</th> <th>Type</th> <th>Join</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>In-House Account Processing</td> <td>Basic Path</td> <td>-</td> </tr> <tr> <td>2a</td> <td>Access 1</td> <td>Alternate</td> <td>6</td> </tr> <tr> <td>4a</td> <td>ATM Path</td> <td>Alternate</td> <td>End</td> </tr> <tr> <td>6a</td> <td>Data Not Found</td> <td>Exception</td> <td>End</td> </tr> </tbody> </table>	Step	Path Name	Type	Join	0	In-House Account Processing	Basic Path	-	2a	Access 1	Alternate	6	4a	ATM Path	Alternate	End	6a	Data Not Found	Exception	End	
Step	Path Name	Type	Join																			
0	In-House Account Processing	Basic Path	-																			
2a	Access 1	Alternate	6																			
4a	ATM Path	Alternate	End																			
6a	Data Not Found	Exception	End																			
	<p>(Only enabled when the basic path is displayed - you cannot add an exception path to another exception path or an alternate pat.)</p> <p>Create a branch from the selected step to an exception path scenario - select the path from the displayed list. If the appropriate scenario does not yet exist, double-click on the <i>new path</i> line and type the scenario name, then click off the line and back on to it. Click on the OK button.</p> <ul style="list-style-type: none"> An entry for this exception path displays in the Entry Points tab, as Step a of the basic path step it branches from; in the Join column, click on the drop-down arrow and select the number of the step at which action flows back to the basic path, or select End if the path terminates separately from the basic path You cannot add more than one branch - whether alternate or exception - to a step You cannot join the end of the branch - whether alternate or exception - to more than one step of the basic path. If you change the join point of the branch, any duplicates of that branch are updated to remain in synchrony with the join point 																					
	Display the Manage Uses Context List dialog, which you use to set the Uses column entries to linked context reference lists	Structured Specification Selected Text Context Menu <small>[675]</small>																				
	Move the currently-selected step one place up (including any <i>new step ...</i> entry)																					
	Move the currently-selected step one place down (including any <i>new step ...</i> entry)																					
	Display a list of diagrams that you can generate from the scenario; select the type of diagram that you want to generate	Generate Diagram <small>[676]</small>																				
	Generate Test Cases based on this Use Case scenario; you can generate either internal Test Cases or External Test Cases	Generate Test Cases <small>[685]</small>																				
	Delete the selected step from the scenario																					
	Display the Help topic for this tab																					

To display the context menu for a structured specification item, right-click on a step or blank line on the **Structured Specification** tab. The following options are available:

Option	Action	Shortcut	See also
Create Structure From Clipboard	Generate a set of steps from a text description or list captured on the clipboard; a new step is		

Option	Action	Shortcut	See also
Text - New Line Delimited Create Structure From Clipboard Text - Sentence Delimited	<p>generated:</p> <ul style="list-style-type: none"> • after each carriage return in the captured text (New Lines), or • for each sentence in the text; that is, after each full stop/space/capital letter combination (Sentences). <p>If a set of steps is already displayed, it is overwritten by the generated steps</p>		
Create Structure From Generated Activity Diagram	<p>Generate a set of steps from an Activity Diagram created for a Use Case</p> <p>If a set of steps is already displayed, it is overwritten by the generated steps</p>		Generate Scenario from Activity Diagram ^[684]
Add Alternate Path	Create a branch from the selected basic path step to an alternate path		Alternate path ^[671]
Add Exception Path	Create a branch from the selected basic path step to an exception path		Exception path ^[672]
Insert Step Above	Insert a <i>new step...</i> line above the currently-selected step (press (Esc) to return this new line to the end of the scenario)	Shift + Insert	
Insert Step Below	Insert a <i>new step...</i> line below the currently-selected step (press (Esc) to return this new line to the end of the scenario)	Insert	
Insert End Step	Insert a <i>new step...</i> line at the end of the scenario	Ctrl + N	
Set Step As 'User' Set Step As 'System'	Switch the entity responsible for performing the action of the selected step between user and system		
Link Step to Use Case	<p>Either include the actions of an existing Use Case element, extend an existing Use Case element, or invoke a Use Case as the action of the selected step</p> <p>Selecting the appropriate sub-option displays the Select Use Case dialog, which you use to browse for and select the required Use Case element</p> <p>The appropriate includes, extends or invokes stereotyped connector is created between the current element and the selected Use Case</p> <p>For the <i>include</i> and <i>extend</i> actions, any text in the Action field is overwritten by the link to the Use Case; for the <i>invoke</i> action, the following link is added to the end of the Action text:</p> <p>(<i>Invokes: <Use Case Name></i>)</p>		Select Use Case ^[690]
Merge With Step	<p>Merge the selected step with another</p> <p>A list of the other steps in the scenario displays; click on the step to merge with the selected step</p>		

Option	Action	Shortcut	See also
Move After Step	Move the selected step to another position in the scenario A list of the other steps in the scenario displays; click on the step after which to position the selected step		
Delete	Delete the selected step; any subsequent steps are moved up one place A prompt displays to confirm the deletion		

To display this context menu, *highlight* the text in a user-editable field within a step on the **Structured Specification** tab. The following options are available:

Option	Action	Shortcut	See also
Create	Create a glossary definition or a new element based on the highlighted text.		Glossary definition ^[769] New element ^[627]
Link Step to Use Case	Either incorporate the actions of an existing Use Case element, extend an existing Use Case element or invoke a Use Case element, as the action of the selected step. Selecting the appropriate sub-option displays the Select Use Case dialog, which you use to browse for and select the required Use Case element. Any text in the Action field is overwritten by the link to the Use Case, except for the invoke action where the following link is added to the end of the Action text: (<i>Invokes: <Use Case Name></i>)		Select Use Case ^[690]
Link to existing Element	(Uses and Results fields only.) Create a Realization or Dependency relationship to a Requirement, Feature or other element elsewhere in the model. You select the element and connector types from submenu options, which then display the Select Element dialog, which you use to browse for and select the required element.		Select Element ^[690]
Insert context reference	Add a reference to an element stored elsewhere in the model, and create an entry for the element in the Context References tab. Selecting this option displays the Select Element dialog, which you use to browse for and select the required reference element.		Context References ^[686] Select Element ^[690] Structured Specification Floating Toolbar ^[676]
Insert glossary definition	Insert an existing glossary term at the cursor position. To select the term, double-click on it in the displayed list. When you select the term it is inserted into the field as highlighted and underlined text, which		

Option	Action	Shortcut	See also
	displays the definition when you move the cursor over it. If you highlighted part of the original text, the term overwrites that text.		
Manage Uses Context List	<p>(On highlighted text or a specific cursor position in the Uses column)</p> <p>Display the Manage Uses Context List dialog, which enables you to:</p> <ul style="list-style-type: none"> • Browse to or type in a context reference item • Re-sort the list with up and down arrow keys • Remove an entry from the list • Enable or disable the Uses Context List facility across the model <p>You can resize the dialog to show the full details of context reference items</p> <p>If you enable the Uses Context List facility, you do not necessarily have to use the dialog to add context list items; you can type or insert an element name from the Context References tab in the Uses column, and that name is added to the Uses Context List</p> <p>If you disable the Uses Context List facility, the context list items in the Uses column are hidden</p>		Context References Tab ^[686]
Split Step	<p>Splits the selected step into two consecutive steps.</p> <p>The option is available only if you highlight a <i>portion</i> of the text in the selected field. The new step takes the highlighted text as its Action text.</p>		
Search for <text>	Displays a sub-menu of options for locating the selected text in a number of locations.		Code editor context menu ^[1402]
Undo	Undo any unsaved changes you have just made in the step.		
Cut	Perform simple editing operations on the highlighted text.		
Copy			
Paste			
Delete			
Select All			

The Entry Points tab shows how the basic path, alternate path and exception path scenarios for the element are organized and interrelated. If an alternate path or exception path has been defined but has not yet been added to the basic path, it is not listed on this tab.

You can switch focus between the Entry Points tab and the Structured Specification tab by pressing (**Alt+Q**) .

To display the context menu for this tab, highlight an entry and right-click on it. The following options are available:

Option	Action	Shortcut	See also
Edit Path	Display the steps of the scenario in the Structured Specification tab, with the first step highlighted		
Join with Step	(Available only if the basic path scenario is displayed in the Structured Specification tab. Not available to edit the basic path scenario.) Highlight the Join field and its drop-down arrow. Click on the drop-down to define or change the step number at which the alternate or exception path rejoins the basic path. Select End if the path does not rejoin the basic path steps.		
Remove Entry Point	(Available only if the basic path scenario is displayed in the Structured Specification tab. Not available to delete the basic path.) Delete the relationship between the selected path and the basic path, and remove the entry from the Entry Points tab.		

Wherever a reference to another element exists on the Scenario tab (that is, where the text is highlighted and underlined), if you hover the cursor over the element name a short floating toolbar displays.

Use To:

- Display the element Properties dialog
- Locate the element in its parent diagram
- Locate the element in the Project Browser

5.6.4.1.8.2 Generate Diagrams

If you have created a structured scenario, you can generate any of the following diagrams from that scenario:

- Activity
 - With ActivityParameter
 - With Action
 - With Action Pin
- Rule Flow
- State Machine
- Sequence
- Robustness

How To:

To generate the required diagram, follow the steps below:

Icon	Step	Action	See Also
	1	Create the scenario structure on the Structured Specification tab	Structured Specification Tab ^[669]
	2	Click on the Generate Diagram icon in the toolbar on the tab	
	3	Click on the type of diagram to generate Enterprise Architect generates the diagram and notifies you that	

Icon	Step	Action	See Also
		<p>generation is complete. Close the Properties dialog to review the diagram</p> <p>If the diagram being generated already exists under the selected element, a prompt displays to overwrite or synchronize with that diagram</p> <p>Select the appropriate radio button to:</p> <ul style="list-style-type: none"> • Overwrite the existing diagram (delete the existing diagram and elements, and create a new diagram and elements) or • Synchronize the elements in the existing diagram with the scenario steps (however, Sequence and Robustness diagrams cannot be synchronized) 	

Notes:

- The **Synchronize elements in existing diagram** option enables the **Preserve Diagram Layout** checkbox, which you can select to preserve the existing arrangement of elements and connectors on the diagram. Any new elements are added to the diagram in the default position, and you manually position them in the diagram as required. If you do not select the checkbox, the diagram is recast in the default layout.

It is recommended that you uncheck the **Preserve Diagram Layout** checkbox if you are synchronizing elements with scenario steps:

- When new steps have been added or existing steps have been deleted or moved within the Use Case
- For the first time in a Use Case that has been imported from XML with the **Strip GUIDs** option selected
- For the first time in a Use Case that has been copied and pasted in the Project Browser, or
- For the first time in a Use Case whose containing package has been copied and pasted in the Project Browser

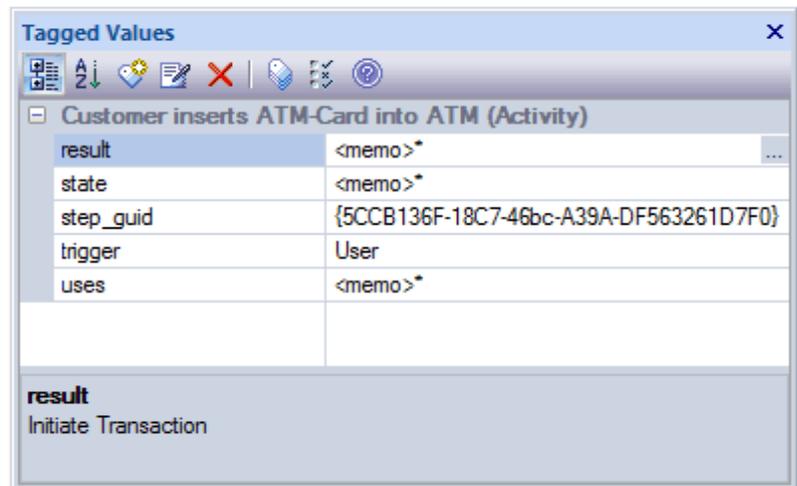
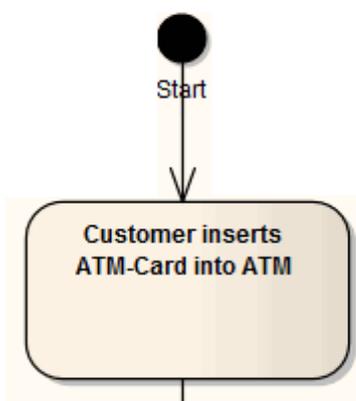
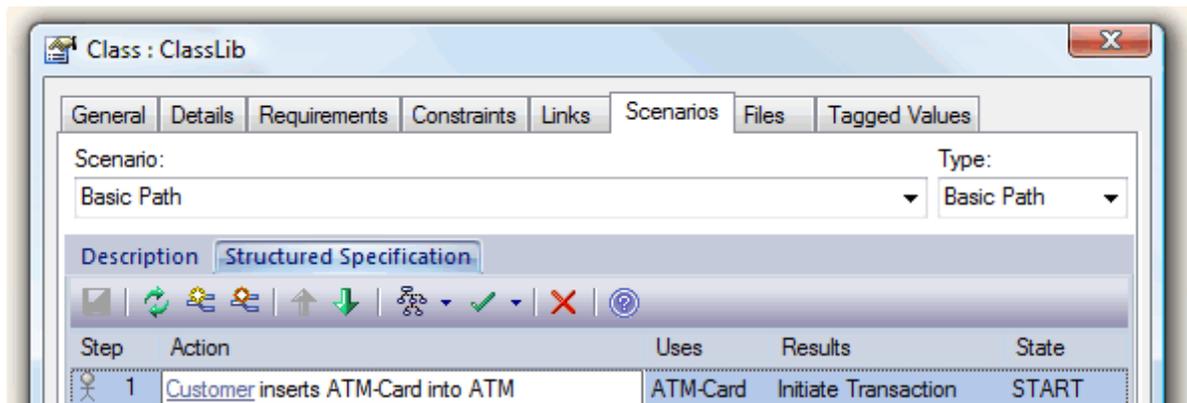
Learn More

- [Generate Activity Diagram](#) [677]
- [ActivityParameter](#) [677]
- [Action](#) [677]
- [Action Pin](#) [677]
- [Generate Rule Flow Diagram](#) [678]
- [Generate State Machine Diagram](#) [679]
- [Generate Sequence Diagram](#) [681]
- [Generate Robustness Diagram](#) [683]
- [Import from XML](#) [323]
- [Copy Element Between Packages](#) [634]
- [Copy A Package](#) [535]

An Activity is generated as a child of the selected element, to act as a container for the diagram.

- The scenario steps are modeled as Activities
- The values of the *Uses*, *Results* and *State* columns for each step are added as Tagged Values of the corresponding Activity

Example:



Topics:

Topic	Detail	See also
Activity with ActivityParameter	<ul style="list-style-type: none"> The values of the Uses and Results columns are modeled as ActivityParameters The value of the State column is added as a Tagged Value of the Activities ActivityParameters are added to the Project Browser and not to the diagram 	
Activity with Action	<ul style="list-style-type: none"> The scenario steps are modeled as Actions The values of the Uses, Results and State columns are added as Tagged Values of the Actions 	
Activity with ActionPin	<ul style="list-style-type: none"> The scenario steps are modeled as Actions The values of the Uses and Results columns are modeled as Input Pins and Output Pins respectively The value of the State column is added as a Tagged Value of the Actions ActionPins are added to the Project Browser and not to the diagram 	

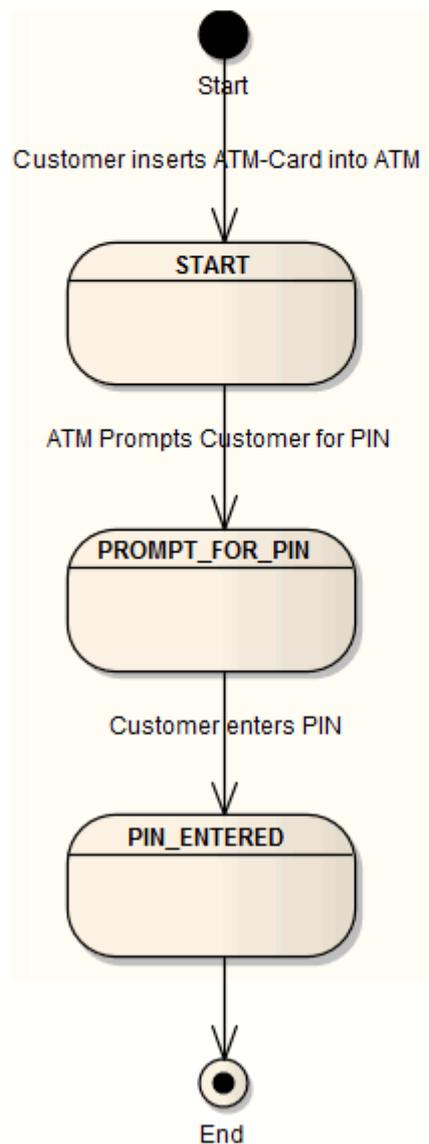
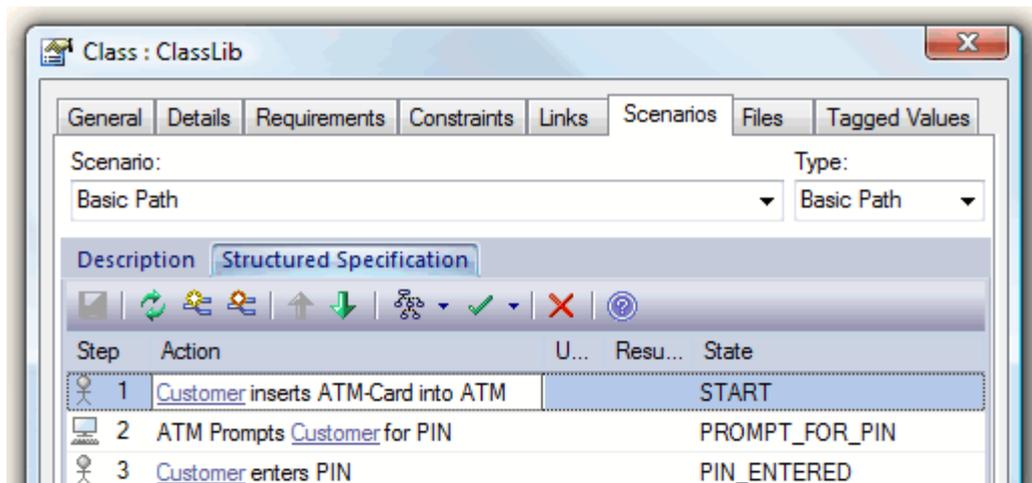
A Rule Flow Activity is created as a child of the selected element, to act as a container for the diagram.

- The scenario steps are modeled as RuleTasks.
- The values of the *Uses*, *Results* and *State* columns are added as Tagged Values of the RuleTasks

A StateMachine is created as a child of the selected element, to act as a container for the diagram.

- Each value in the *State* column is modeled as a State.
- The scenario steps become the Transition connectors between the States
- The values of the *Uses* and *Results* columns are added as Tagged Values of the Transitions.

Example:

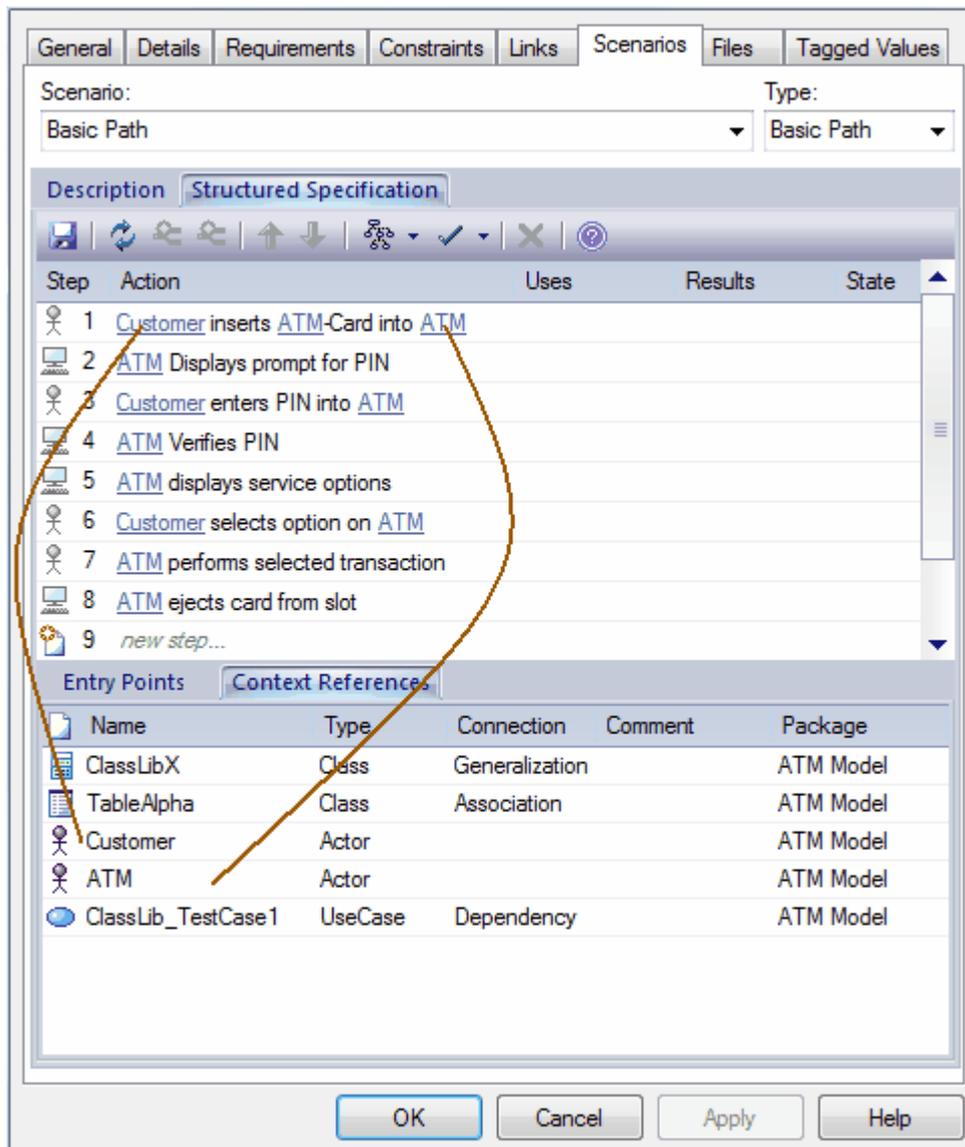


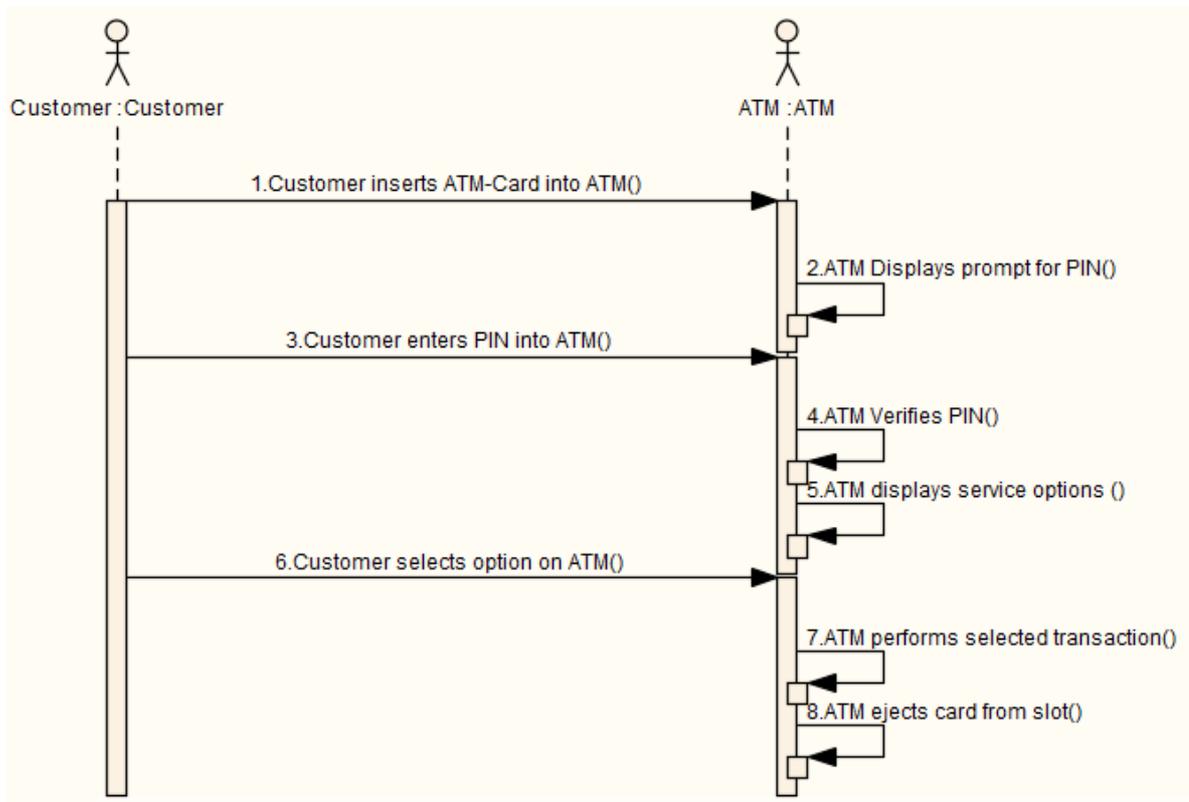
An Interaction is created as a child of the selected element, to act as a container for the diagrams - the Basic, Alternate and Exception paths are modeled as separate Sequence diagrams under the Interaction.

Topics:

Topic	Detail	See also
Usage	<p>All the elements involved in the scenario should be identified in the Context Reference tab. That is, relationships must already exist between the scenario parent element and the other elements named in the scenario.</p> <ul style="list-style-type: none"> • Each Context Reference element named in a scenario step is modeled as a Lifeline • The step itself becomes the Message between an originator and its destination(s) • The first Context Reference element in a step is treated as the originator • The subsequent Context Reference element(s) become the destination(s) <p>Because the diagram generator acts on element names in the step, you should take care to avoid using the element names as normal text. For example, in step 1 in the dialog below, the term ATM-Card is interpreted as a reference to the ATM element, and two Customer inserts ATM-Card into ATM Messages are generated for the step. (To avoid confusion, in the diagram the first Message has been deleted.)</p> <p>If the Show Sequence Numbering checkbox on the Diagram Sequence page of the Options dialog is not selected, the Message name on the Sequence diagram is prefixed by the step number, as shown in the diagram below. If the checkbox is selected, the message sequence number is shown instead of the step number</p> <p>The values of the Uses, Results and State columns are added as Tagged Values of the Message</p>	<p>Context Reference Tab⁶⁸⁶</p> <p>Sequence⁴³¹</p>

Example:

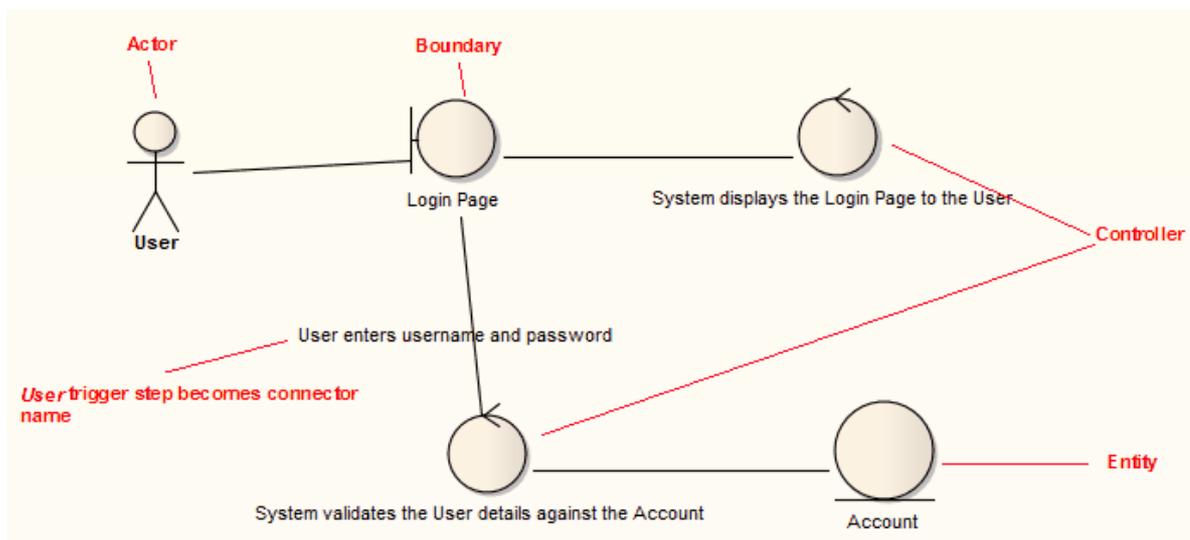




A Collaboration is created as a child of the selected element, to act as a container for the Robustness diagram.

Example:

Step	Action
1	System displays the Login Page to the User
2	User enters username and password
3	System validates the User details against the Account



Notes:

- All the elements involved in the scenario should be identified in the Context Reference tab. That is, relationships must already exist between the scenario parent element and the other elements named in the scenario
- Any values in the *Uses*, *Results* and *State* columns are ignored and not represented in the diagram
- Each UI element in a step becomes a Boundary element. A Dependency relationship is created from this Boundary element to the UI element (this connector is not shown on the diagram)
- Each Actor referenced in a step is dropped into the Robustness diagram as a simple link
- Each Class referenced in a step is dropped into the Robustness diagram as a simple link, and is given the stereotype *entity*
- Each step with a *System* trigger becomes a Controller. Alternate/exception path Controllers are displayed with a red background color
- Each step with a *User* trigger becomes the name of the Association between Controllers

Learn More:

- [Collaboration](#)^[945]
- [Robustness Diagram](#)^[1186]
- [Context Reference Tab](#)^[686]
- [UI Elements](#)^[1282]
- [Boundary Elements](#)^[1288]
- [Dependency](#)^[979]
- [Control](#)^[1289]
- [Associate](#)^[970]

5.6.4.1.8.3 Generate Scenario From Activity Diagram

You can generate a range of diagrams from a scenario in an element. Conversely, you can also generate a structured scenario within an element from an Activity diagram, reverse engineering the steps from the diagram elements (effectively either regenerating the scenario within the Use Case, or transferring a scenario into another Use Case).

How To:

To generate the scenario from the Activity diagram, follow the steps below:

Step	Action	See Also
1	Open the element Properties dialog, select the Scenarios tab, and select the Structured Specification tab	
2	Right-click in the empty space within the tab, and select the Create Structure From Generated Activity Diagram context menu option The Select an Activity (generated from a UseCase Scenario) containing the Diagram dialog displays	Select <Item> Dialog ^[690]
3	Search for and select the Activity containing the required diagram. Enterprise Architect validates the diagram (displaying the results in the Output window) and, if the diagram is valid, generates the scenario steps in the Structured Specification tab (replacing any existing scenario steps).	Output Window ^[127]

Notes:

- The source Activity diagram must be generated from another Use Case Scenario
- Any existing scenario steps are deleted and replaced by the generated scenario
- This facility does not operate on the enhanced Activity diagrams generated from a Use Case - those generated with ActivityParameters, Actions and Action Pins

Learn More:

- [Generate Diagrams](#) ^[676]

5.6.4.1.8.4 Generate Test Cases

When you select a scenario and click on the **Test Case Generation** icon in the window toolbar, Enterprise Architect prompts you to select to generate either an Internal Test Case or an External Test Case.

Topics:

Topic	Detail	See also
Internal Test Case	<p>A test is generated for the basic path and each alternate and exception path in the scenario, and added to the selected element</p> <p>In addition, for each step in the basic, alternate and exception paths that has a value in the Results column, a test is generated and added to the selected element</p> <p>A diagram is created under the selected element, to which the selected element and a Note (showing the element's Structured Specification) is added</p> <p>The created tests are displayed in the <i>Test Scripts</i> compartment of the selected element on the diagram</p> <p>Another way to view these tests is to click on the element and display the Testing window</p> <p>These generated tests are written to the Scenario test tab of the Testing window; you can move the tests to another test-type tab if required</p>	<p>Show Test Scripts compartment ^[1707]</p> <p>The Testing Window ^[1695]</p> <p>Move Or Copy Tests Between Categories ^[1703]</p>
External Test Case	A Test Case element is created, linked to the selected element using the Trace connector	Test Case ^[1674]

Topic	Detail	See also
	<p>A diagram is created under this Test Case element, to which the selected element, the Test Case element and a Note (showing the element's Structured Specification) is added</p> <p>The created tests are displayed in the <i>Test Scripts</i> compartment of the Test Case element on the diagram</p> <p>A test case is generated and added to the Test Case element for the basic path, and for each alternate and exception path</p> <p>In addition, for each step in the basic, alternate and exception paths that has a value in the Results column, a test is generated and added to the Test Case element</p> <p>You can review the tests within the Test Case element using the Testing window, as for Internal Test Cases</p>	Trace ^[1012]

5.6.4.1.8.5 Context References Tab

On the Scenarios page of the element Properties dialog, the Context References tab displays a list of all elements that are either:

- Connected to the current element by any connector, on the current diagram or another, or
- Defined as a cross reference (or custom reference) on the current element

Topics:

Topic	Detail	See also
Usage	<p>This page enables you to add custom references - right-click anywhere in the list and select the Add Reference context menu option</p> <p>The Select Element dialog displays, in which you can locate and select the required cross reference element or elements</p> <p>For each cross reference you can also use context menu options to delete the entry in the list, or to open the Comment field so that you can add or edit comment text</p> <p>For each element in the Context References list, wherever the name of that element appears in the structured specification, the name is highlighted and underlined; press (Ctrl) + click on the highlighted name to view the element Properties dialog</p>	Select <Item> Dialog ^[690]

Learn More:

- [Custom/Cross Reference](#)^[630]

5.6.4.1.8.6 Scenario Constraints Tab

The Structured Specification Constraints tab is a simple link to the Constraints tab of the element overall. It lists existing constraints; if you select the toolbar icons to add or edit a constraint, control switches to the element Constraints tab.

Learn More:

- [Constraints](#)^[666]

5.6.4.1.9 Associated Files

An element can be linked to files held in the database, using the Files page of the element's Properties dialog.

Field/Button	Usage	See Also
File Path	Specifies the directory path and name of the file	
Type	Display the local file or web address	
Last Write	Display the date and time the file was last updated	
Size	Display the size of the file	
Notes	Indicates extra information about the file	
Files	Display the list of linked files	
Launch	Open the selected file Local files open with their default application and web files open in the default browser	

Notes:

- Linked files are a good way to link elements to additional documentation and/or source code
- You can also insert hyperlinks in diagrams to other files, and launch them directly from the diagram

Learn More:

- [Hyperlinks](#) ^[1292]

5.6.4.2 The Element Browser

The Element Browser window lists a range of added-on properties of the selected element.

Access: [View](#) | [More Element Tools](#) | [Element Browser](#)

Topics:

Topic	Detail	See also
Properties	The following properties are listed, where they are present in the element: <ul style="list-style-type: none"> • Operations • Attributes • Linked Features (Linked Operations and Attributes) • Tagged Values • Constraints • Internal Requirements (Responsibilities) • Attached Files • Relationships • Scenarios • Maintenance Items 	Operations ^[707] Attributes ^[695] Linked Features (Linked Operations and Attributes) ^[742] Tagged Values ^[762] Constraints ^[666] Internal Requirements ^[664] (Responsibilities)

	<ul style="list-style-type: none"> • Testing Items • Project Management Items 	Attached Files ^[667] Relationships ^[666] Scenarios ^[667] Maintenance Items ^[1713] Testing Items ^[1695] Project Management Items ^[346]
--	---	--

Notes:

- If you double-click on the element name at the top of the dialog (the root node) the Properties dialog for the element displays, at the General tab
- If you right-click on a folder name, the **Edit <object>** context menu option displays; when you select this option, the appropriate window or dialog displays, or the element Properties dialog displays the appropriate tab, to enable you to update the items in the folder
- If you double-click on an item within the *Requirements*, *Scenarios* or *Constraints* folders, the docked window or view displays with the focus on the selected item
- If you double-click on an item within the *Relationships* folder, the Properties dialog displays for that relationship
- You can drag a relationship from the Element Browser onto a diagram; if the element at one or both ends of the relationship is not already on the diagram, the element is also added to the diagram, and if the relationship is already on the diagram, but hidden, it becomes visible again
- Instead of dragging the relationship onto the diagram, you can right-click on it and select context menu options to place the relationship and elements on the diagram, and to show or hide the relationship
- If you double-click on an item within the *Files* folder, the file opens either on a separate tab in the Diagram View workspace (if the file can be opened within Enterprise Architect) or in the default Windows viewer/editor for the file type (if the file cannot be opened within Enterprise Architect)
- If you double-click on an operation or attribute in the *Operations* or *Attributes* folders, the appropriate Properties dialog displays
- If you right-click on an attribute or operation, whether in the *Attributes* or *Operations* folder or in the *Linked Features* folder, a context menu displays that enables you to display the source code; alternatively, click on the attribute or operation and press (**Ctrl+E**), or - in the *Linked Features* folder - double-click on the attribute or operation
- The source code viewer in which the source code displays depends on which editor you select as the default, either for the project as a whole or for a specific programming language; if you select the Enterprise Architect internal editor, the code displays in the Source Code Viewer with the cursor positioned on the selected feature
- The right-click context menu for an operation also provides options to set a number of types of recording marker

Learn More:

- [Scenarios and Requirements Window](#) ^[689]
- [Connector Properties](#) ^[756]
- [Options - Code Editors](#) ^[1519]
- [Source Code Viewer](#) ^[1408]
- [Recording Markers](#) ^[1642]

5.6.4.3 Properties Window

The Properties window provides a convenient way to view (and in some cases edit) common properties of elements. When an element is selected, the Properties window shows the element's name, stereotype, version, author, dates and other pertinent information.

Access: [View | Element Properties](#)

Topics:

Topic	Detail	See also
Properties Sections	The Properties window is divided into three expandable sections: <ul style="list-style-type: none"> • <Element type> Settings - for the basic element details • Project - for general housekeeping settings • Advanced - only active for generalizable elements 	

Notes:

- When you click on a field name, a brief explanation of that field displays at the bottom of the Properties window, unless you have selected the **Hide Properties Info Section** checkbox on the General page of the Options dialog
- If you click on the field value for an editable field, a drop-down arrow displays that enables you to select a different value

Learn More:

- [Hide Properties Info Section](#)^[423]

5.6.4.4 The Scenarios & Requirements Window

The Scenarios & Requirements window provides a convenient way to quickly add, view, edit and delete rules applied to an element. The window shows details of the entities that impose such rules or restrictions on the element, namely:

- The element's internal responsibilities or requirements
- The element's internal constraints (not external Constraint notes)
- The element's scenarios

Access: [View | More Element Tools | Scenarios & Requirements \(Ctrl + Shift + 3\)](#)

Topics:

Topic	Detail	See also
Usage	The Scenarios & Requirements window is typically used to examine Use Case and Test Case elements, and any other elements that realize an external Requirement For convenience, you can display the window as either a dockable window around the edge of your workspace, or as a view in the center of your workspace (in which the Scenarios, Constraints and Requirements are shown on separate tabs)	Notes ^[913]

	Use the  icon in the window toolbar to toggle between these display options	
--	--	--

How To:

To review an internal requirement, constraint or scenario for an element, follow the steps below:

Step	Action	See Also
1	Select the element in the Project Browser or diagram and select the Scenarios & Requirements menu option The Scenarios & Requirements window displays	
2	Either: <ul style="list-style-type: none"> Click on the required item in the list panel on the left of the window, or If the Element Browser is not already displayed, click on the  icon and select the required item from the <i>Constraints</i>, <i>Requirements</i> or <i>Scenarios</i> folder The appropriate screen displays, with the details of the selected item in the fields	Element Browser ^[687]
3	The toolbar icons, from left to right, enable you to: <ul style="list-style-type: none"> Add a new rule to the current screen (click on the appropriate folder to create an item of a different type) Save the new or edited item Save the current (edited) item as a new item Delete the current item Sort the contents of the selected folder into alphabetical order (Print icon unavailable) Display the Element Browser window, or highlight the current item in the Element Browser window Switch between a docked window display and a workspace view display You can also add or edit formatted notes in the Notes field, using the Notes toolbar at the top of the field The list panel also provides a context menu that provides options for adding a new item to a folder, deleting the selected item or, for a requirement, converting an internal responsibility into an external Requirement element	Notes Toolbar ^[770] Converting an Internal Responsibility Into an External Requirement ^[1167] Scenarios ^[667] Constraints ^[666] Requirements ^[664]

5.6.4.5 Select <Item> Dialog

The Select <Item> dialog is a multi-purpose browser and search tool for locating model items such as Classifier elements, properties, attributes and behaviors. The <Item> in the dialog title changes to represent the type of item the original operation is working on.

Topics:

Topic	Detail	See also
Usage	The dialog is called in a range of operations; for example, setting: <ul style="list-style-type: none"> The base type or classifier for an Object, Swimlane or 	Using Classifier ^[706] Objects ^[957]

	<p>Lifeline</p> <ul style="list-style-type: none"> • Classifiers for the return types for operations • Classifiers for generalization sets • The associated behavior for a behavior call • The type and return type for operation parameters • Activities for State transitions • Use Case Scenarios • Pattern element defaults • Activities from which to generate code • The values of Tagged Values 	<p>Swimlanes ^[590]</p> <p>Lifeline ^[910]</p> <p>Return Types ^[708]</p> <p>Generalization Sets ^[750]</p> <p>Associate with Different Behaviour ^[716]</p> <p>Insert Operation Parameters ^[726]</p> <p>Transitions ^[1013]</p> <p>Generate Scenarios from Activity Diagram ^[684]</p> <p>Use a Pattern ^[1024]</p> <p>Assigning Tagged Values ^[764]</p>
<p>Multiple Selection</p>	<p>Where an operation permits the selection of multiple items, the Select <Item> dialog is automatically enabled to support this; to select the items, press (Ctrl) as you click on each item</p> <p>Having selected an item, you can continue to expand and browse the hierarchy, and/or search for items; the dialog retains the existing selections until you click on the OK button</p>	

How To:

To select a required item, follow the steps below:

Step	Action	See Also
1	<p>During an operation, when it is necessary to locate an element or feature, click on the (...) (browse) button</p> <p>The Select <Item> dialog displays</p>	
2	<p>If required, in the Go To Namespace field select a namespace to reduce the scope of the displayed hierarchy</p> <p>The dialog opens the section of the hierarchy associated with that namespace, and closes all previously-open sections associated with other namespaces</p>	
3	<p>You can either:</p> <ul style="list-style-type: none"> • Expand the selected area of the hierarchy on the Browse tab, or any other package, and locate the required item (go to step 5) or • Click on the Search tab and, in the Find field, type a partial or complete text string to search for the item 	
4	<p>On the Search tab, you can filter the search further by selecting the Whole Word and Match Case checkboxes</p> <p>Each list entry shows the name of the item, the type, any stereotype the item has,</p>	

Step	Action	See Also
	<p>the immediate package in which the item is held, and any successive parent packages (the package path)</p> <p>You can either:</p> <ul style="list-style-type: none"> • Select the item immediately on the Search tab or • Right-click on one item or a group of items and select the Locate item(s) in tree context menu option; this redisplay the Browse tab and highlights each selected item in the <namespace> hierarchy 	
5	Click on the required item	
6	Click on the OK button	

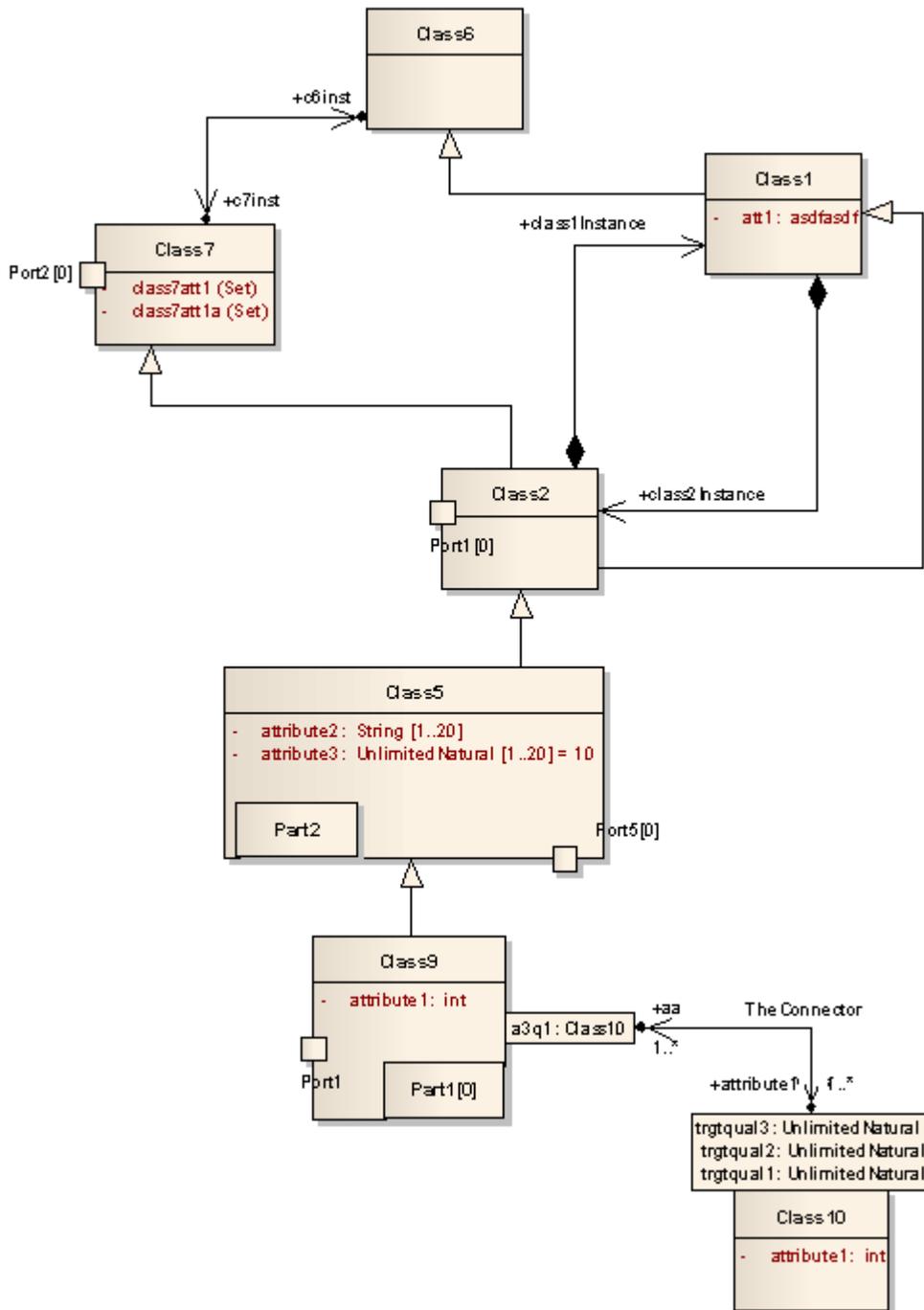
Notes:

- When you have selected an item, the Select <Item> dialog retains the context and item; next time you display the dialog, if the context is similar the dialog opens to the same Namespace and item - for example, if you have selected an activity for a State transition and you start to do the same for another transition, the dialog opens to the activity you previously selected
- If the context is totally dissimilar, the dialog opens with the Namespace <any> and a collapsed model hierarchy
- If the available items do not meet your requirements, you can create a new item and define the appropriate properties - click on the **Add New** button; the appropriate Add <Item> dialog displays, on which you define the required item
- The **Add New** button is not always available, depending on the context and the type of item being searched for

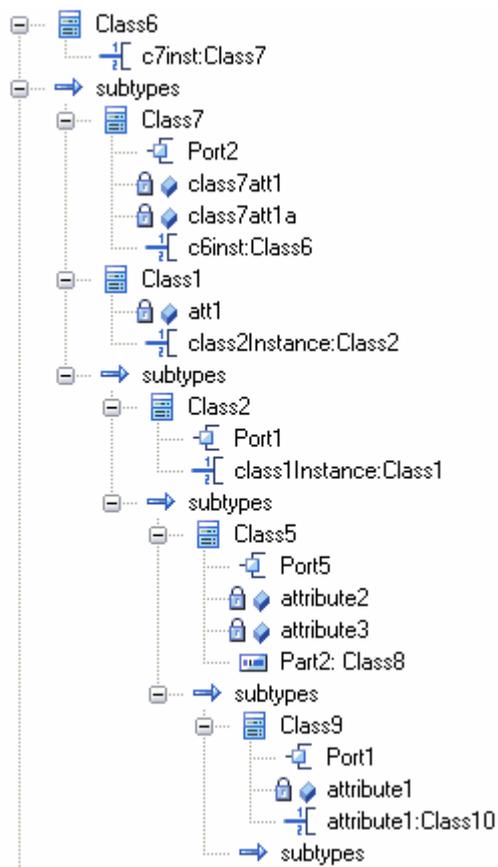
5.6.4.5.1 Select Property Dialog

The Select Property dialog is a specific instance of the Select <Item> dialog; it is used to select Ports, Parts, Attributes and Association Ends as *redefined* or *subsetting* properties, from a *hierarchical list* of Classes and their contained properties in the model. For example, consider the section of the model below:

Example:



This would be represented in the dialog as follows:



Locate and click on the required object to select it, then click on the **OK** button (to select several objects at once, press and hold (**Ctrl**) while you click on each object).

Notes:

- AssociationEnds should be owned by the Class to be listed in the dialog

Learn More:

- [Select <Item> Dialog](#)^[690]

5.6.4.5.2 Set Feature Dialog

The Set Feature dialog is the Set Operation dialog used to change the operation represented by an Action on an Activity diagram.

As the Set Operation or Set Attribute dialog, it is also used to set the *Value* operation or attribute for Tagged Values of type RefGUID or for the target of a hyperlink in a diagram.

How To:

To locate and select a value operation or attribute, follow the steps below:

Step	Action	See Also
1	The Set Operation (or Set Attribute) dialog displays, with the model hierarchy	

Step	Action	See Also
	opened at the point at which you selected the original operation or attribute	
2	If required, in the In Namespace field, click on the drop-down arrow and select another model that contains the required operation or attribute The package hierarchy for that model displays	
3	Browse through the hierarchy, or use the Search tab to locate the required operation or attribute, then double-click on the item to select it	Select <Item> Dialog ^[690]

Learn More:

- [Class Operations In Activity Diagrams](#)^[872]
- [Pre-defined Structure Types - RefGUID](#)^[1110]
- [Hyperlinks](#)^[1292]

5.6.5 Attributes

Attributes are features of an element that represent the properties or internal data elements of that element. Not all element types support attributes, and others have restrictions - for example, attributes of Interfaces must have Public scope.

Elements with attributes (typically Classes) display their features in diagrams in the manner illustrated below. Attributes display in the first compartment of properties in colored text - the default color is red (for example, *Age : int*).

Example:

For a Customer Class, **CustomerName** and **CustomerAddress** can be attributes. Attributes have several important characteristics, such as **type**, **scope** (visibility), **static**, **derived** and **notes**.



Topics:

Topic	Detail	See also
Create and Modify Element Attributes	<p>In the Project Browser, Classes with attributes have their features collected beneath them, each preceded by a blue box ()</p> <p>In the Diagram view, either:</p> <ul style="list-style-type: none"> • Right-click on the element to be edited, and from the context menu select the Attributes menu option • Click on the element and press (F9), or • Drag the attribute from the Project Browser onto the element <p>The <Element name> Attributes dialog displays</p> <p>This facility is only available if the element supports attributes</p>	

Notes:

- If you make changes and do not save them, the **Cancel** button prompts you to save or cancel the changes, whilst the **Close** button closes the dialog immediately and does not save the changes
- If you are creating many attributes, go to the Attribute/Operations page of the Options dialog (**Tools | Options | Source Code Engineering | Attribute/Operations**) and select the **After save, re-select edited item** checkbox; now, when you create an attribute and click on the **Save** button, the dialog fields clear ready for you to enter the details of the next attribute - this helps you when you want to create attributes quickly and might not necessarily want to fully define each one as you create it
- If the parent element provides source or target roles for a connector, the connector can be attached to a specific attribute (feature)

Learn More:

- [Attribute Tagged Values](#) ^[699]
- [Create Properties](#) ^[700]
- [Display Inherited Attributes](#) ^[702]
- [Create Object From Attribute](#) ^[704]
- [Attributes Dialog - General](#) ^[696]
- [Attributes Dialog - Detail](#) ^[698]
- [Attributes Dialog - Constraints](#) ^[699]
- [Connect to Element Feature](#) ^[742]

5.6.5.1 Attributes Dialog - General

This topic describes the fields and options of the General page of the Attributes dialog.

Reference:

Field	Usage	See also
Name	Display the name of the attribute; for a new attribute, type the name (with no spaces)	
Type	Display the attribute's type; if necessary, either: <ul style="list-style-type: none"> • Click on the drop-down arrow and select a different type, or 	Data Types ^[777] Instance

Field	Usage	See also
	<ul style="list-style-type: none"> Click on the (...) button to open the Select <Item> dialog and select or define a different attribute classifier type that might not be in the Type drop-down list <p>The type can be defined by the code language (data type) or by a classifier element; when you click on the drop-down arrow, the first set of values in the list provides the data types, and the second (longer) set provides the possible classifiers</p> <p>To add new code language data types that can be displayed in this list, see the <i>Data Types</i> topic</p>	Classifier ⁶⁹⁰
Initial Value	Display an optional initial value; if necessary, type in or browse for a new initial value	
Stereotype	Define the optional stereotype of the attribute; if necessary, either: <ul style="list-style-type: none"> Type a different stereotype name Click on the drop-down arrow and select a stereotype, or Click on the (...) button and browse for the stereotype 	
Alias	Display an optional alias for the attribute; if necessary, type in a new alias	
Scope	Define the attribute as Public , Protected , Private or Package ; if necessary, click on the drop-down arrow and select a different scope	
Static	Indicate that the attribute is a static member	
Const	Indicate that the attribute is a constant	
Is Literal	(For Enumeration elements) Defaults to selected, to define the attribute as an enumeration literal Deselect to define the attribute as a normal element attribute In the <i>Attributes</i> compartment on the diagram, the enumeration literals are listed separately, above the normal attributes (ensure that the Stereotype field for the normal attribute is not set to enum)	
Derived	Indicate that the attribute is derived from one or more other attributes; for example, full name can be derived from the <i>last name</i> and <i>first name</i> attributes	
Containment	Define the containment type (By Reference , By Value or Not Specified); if necessary, click on the drop-down arrow and select a different containment type	
Property	Indicate that the attribute has automatic property creation; when you select this checkbox, the Create Property Implementation dialog displays	Create Properties ⁷⁰⁰
Notes	Enter any free text notes associated with the attribute; you can format the notes text using the Notes toolbar at the top of the field	Notes Toolbar ⁷⁷⁰

Topics:

Topic	Detail	See also
Usage	To change the position of a selected attribute in the list at the bottom of the dialog, click on the Scroll Up or Scroll Down (hand) buttons	

Topic	Detail	See also
	<p>To review an existing attribute, click on the attribute name in the list</p> <p>To delete an existing attribute, click on the attribute name in the list and click on the Delete button</p> <p>To create a new attribute, either:</p> <ul style="list-style-type: none"> • Click on the New button, or • Click on an existing attribute name in the list, and click on the Copy button <p>If you have changed the attribute details, click on the Save button to save the changes</p>	

Notes:

- By default, the attributes are listed in alphabetical order; before changing this sequence, you must deselect the **Sort Features Alphabetically** checkbox on the Objects page of the Options dialog (**Tools | Options | Objects**)

5.6.5.2 Attributes Dialog - Detail

This topic describes the fields and options of the Detail page of the Attributes dialog.

Reference:

Field	Action	See also
Lower bound	Define a lower limit to the number of elements allowed in the collection	
Upper bound	Define an upper limit to the number of elements allowed in the collection	
Allow Duplicates	Indicate that duplicates are allowed Maps to the UML property isUnique, value FALSE	
Ordered Multiplicity	Indicate that the collection is ordered	
Redefined Property	Review the redefined properties for the attribute Add redefined properties by clicking on the Add button to display the Select Property dialog	Select Property dialog ^[692]
Subsetted Property	Review the subsetted properties for the attribute Add subsetted properties by clicking on the Add button to display the Select Property dialog	
Collection	Code the attribute as an array, so that it can contain multiple concurrent values rather than a single value	
Attribute is a Collection	Indicate that the attribute is a collection (array)	
Container Type	Enter the name of the container type	
Transient	(For Java code) indicate that the attribute can change	

Field	Action	See also
	regardless of what the code is performing	
Qualifiers	Click on this button to add Qualifiers to the attribute The Qualifiers dialog displays	Qualifiers ^[971] Qualifiers dialog ^[972]

5.6.5.3 Attributes Dialog - Constraints

Attributes can have constraints associated with them; typically these indicate such things as maximum value, minimum value and length of field.

These constraints are managed on the Constraints page of the Attributes dialog.

Topics:

Topic	Detail	See also
Usage	Select the Constraints page to define the constraints <ul style="list-style-type: none"> To review an existing constraint, click on the constraint name in the panel at the bottom of the dialog To delete an existing constraint, click on the constraint name in the panel and click on the Delete button To create a new constraint, click on the New button Review, edit or complete the fields as indicated in the table below If you have created or edited the data, click on the Save button to save the changes	

Reference:

Field	Usage	See also
Constraint	Type in the constraint name	
Type	Type in the constraint type	
(Notes)	Type in any comments or notes concerning the constraint	

5.6.5.4 Attribute Tagged Values

You can define Tagged Values for an attribute. Tagged Values are a convenient means of extending the properties a model element supports. This in turn can be used by code generators and other utilities to transform UML models into other forms.

How To:

To add a Tagged Value to an attribute, follow the steps below:

Step	Action	See Also
1	Click on the Tagged Values tab of the Attribute Properties dialog	

Step	Action	See Also
2	Click on the New tag button () The Tagged Value dialog displays	
3	In the Tag field, type the tag name or click on the drop-down arrow and select a defined tag	
4	If appropriate, in the Value field type a specific value for the tag	
5	Click on the OK button to confirm the operation The tag name and value are displayed under the attribute in the Tagged Values tab	

Notes:

- Tagged Values are supported for attributes, operations, objects and connectors
- You can define custom tags by creating a Custom Tagged Value Type

Learn More:

- [Creating a Custom Tagged Value Type](#) ^[1113]

5.6.5.5 Create Properties

Enterprise Architect has capabilities for automatically creating properties in various languages. Property creation is controlled from the General page of the Attributes dialog.

Topics:

Topic	Detail	See also
Usage	<p>On the General page, select the Property checkbox</p> <p>The Create Property Implementation dialog immediately displays</p> <p>The Language panel defaults to the Class language; however, you can change this and generate the properties for any language, including Action Script</p> <p>Each language has slightly different syntax and generates slightly different results; for example:</p> <ul style="list-style-type: none"> • Java and C++ generate get and set functions • C# and VB.Net create property functions • Delphi creates get and set functions as well as a specialized Delphi property Tagged Value <p>Type in the required details and click on the OK button</p> <p>Enterprise Architect generates the required operations and properties to comply with the selected language.</p> <p>Note that get and set functions are stereotypes with «<i>property get</i>» «<i>property set</i>» making it easy to recognize property functions; you can also hide these specialized functions by deselecting the Property Methods checkbox in the Features tab of the Diagram Properties dialog for a specific diagram (select the Diagram Properties menu option) - this makes it</p>	Compartments ^[659]

Topic	Detail	See also
	<p>easier to view a Class, uncluttered by many <i>get</i> and <i>set</i> methods</p> <p>Note that for Delphi you must enable the Tagged Values compartment to see the generated properties</p>	

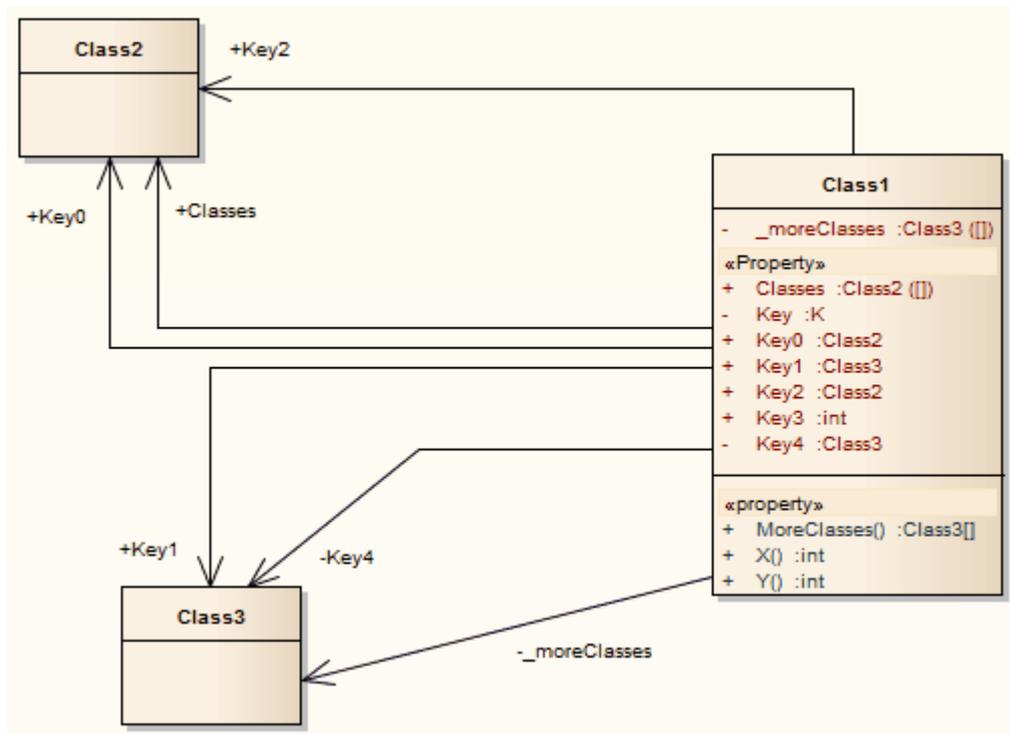
5.6.5.6 C# Automatic Property Support

For C# at and beyond release 4.0, Enterprise Architect enables you to model *automatic properties* as «*Property*» stereotyped attributes.

Topics:

Topic	Detail	See also
Properties of Attributes	<p>When you create the attribute, on the General page of the Attributes dialog simply set the Stereotype field to Property; the generated code resembles the following:</p> <pre>public class Class1 { public Class2() Classes {get; set;} }</pre> <p>This is in contrast to the Enterprise Architect facilities for modeling get/set methods as properties of an attribute in various other languages, generating code such as:</p> <pre>public class Class1 { public Class3() m_delivery; : public Class3() delivery { get{ } set{ } } }</pre> <p>You cannot create properties with get/set methods on an automatic properties attribute; whilst the «<i>Property</i>» stereotype is assigned in the Attributes dialog, the Property checkbox is grayed out</p>	<p>Create Properties </p>
Tagged Values	<p>You can also add the following Tagged Values to automatic properties to denote specific characteristics, if required:</p> <ul style="list-style-type: none"> • writeonly - the property only has a <i>set method</i>; e.g. <code>public int Prop1 { set; }</code> • readonly - the property only has a <i>get method</i>; e.g. <code>public int Prop2 { get; }</code> • GetScope - specifies the property's <i>Get Scope</i>; e.g. <code>public int Prop3 { protected get; set; }</code> 	

Topic	Detail	See also
	<ul style="list-style-type: none"> SetScope - specifies the property's <i>Set Scope</i>; e.g. <code>public int Prop4 { get; protected set; }</code> 	
Reverse Engineering	Imported automatic properties generate attributes with a « <i>property</i> » stereotype and appropriate Tagged Values, and - if the property type is a recognized element in the model - an Association connector	

Example:**5.6.5.7 Display Inherited Attributes**

When displaying a Class with attributes in a diagram, you can also show the inherited attributes from all parents in the elements type hierarchy (ancestors).

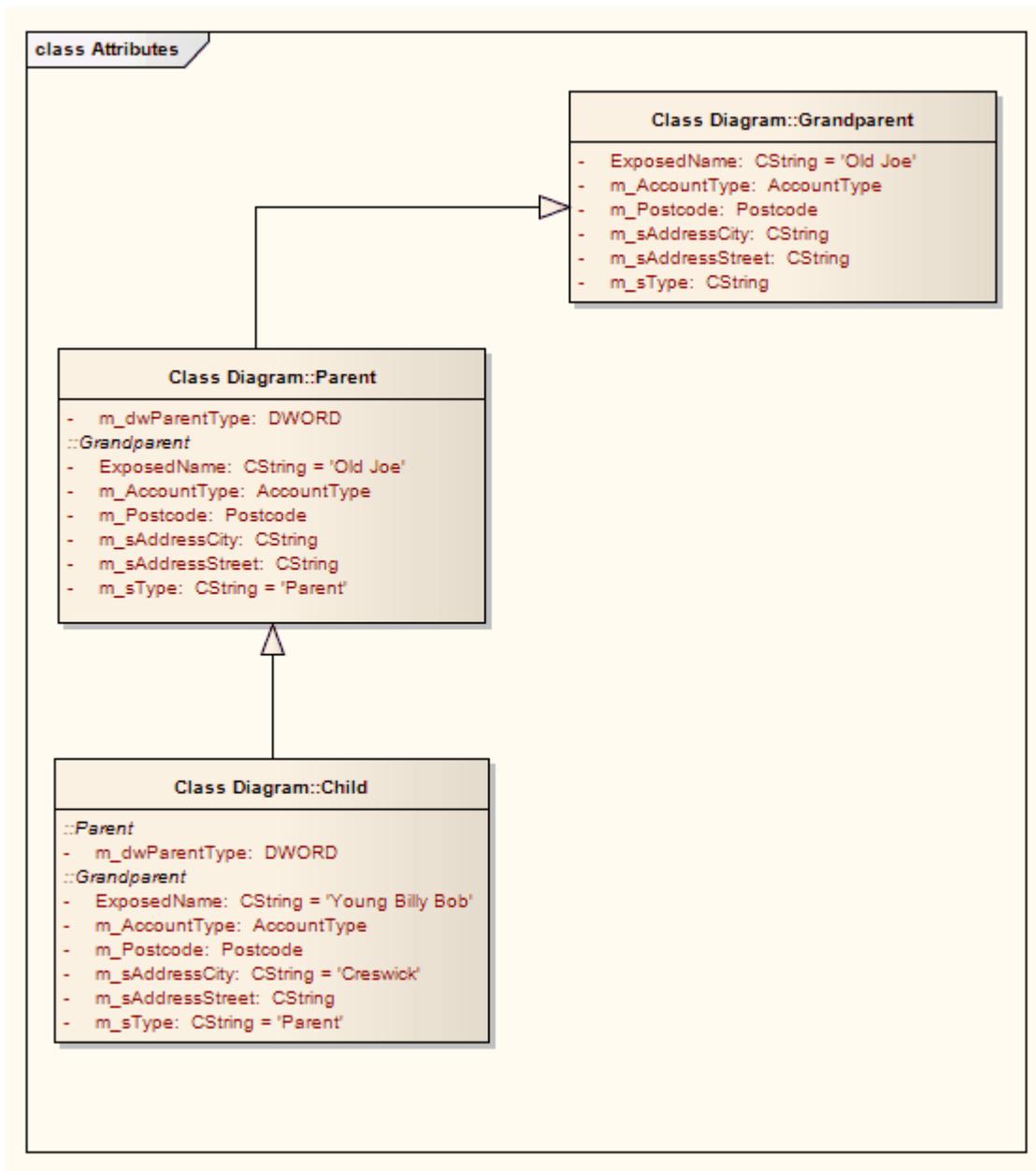
Access: **Right-Click element (on diagram) context menu | Feature Visibility (Ctrl + Shift + Y)**

Topics:

Topic	Detail	See also
Override Attribute Initializers	<p>Note that for elements that have attributes, you can also override an inherited attribute's initial value using the element context menu option Advanced Override Attribute Initializers to display the Override Attribute Initializers dialog</p> <p>In this dialog, select the variable name and enter a new initial value</p>	

Topic	Detail	See also
	<p>If required, type a note in the Note field</p> <p>When you display inherited attributes, Enterprise Architect merges the list of attributes from all ancestors and merges the attribute initializers, so that the final child Class displays the correct attribute set and initial values</p>	

Example:



Learn More:

- [Feature Visibility](#) ⁵⁸⁵

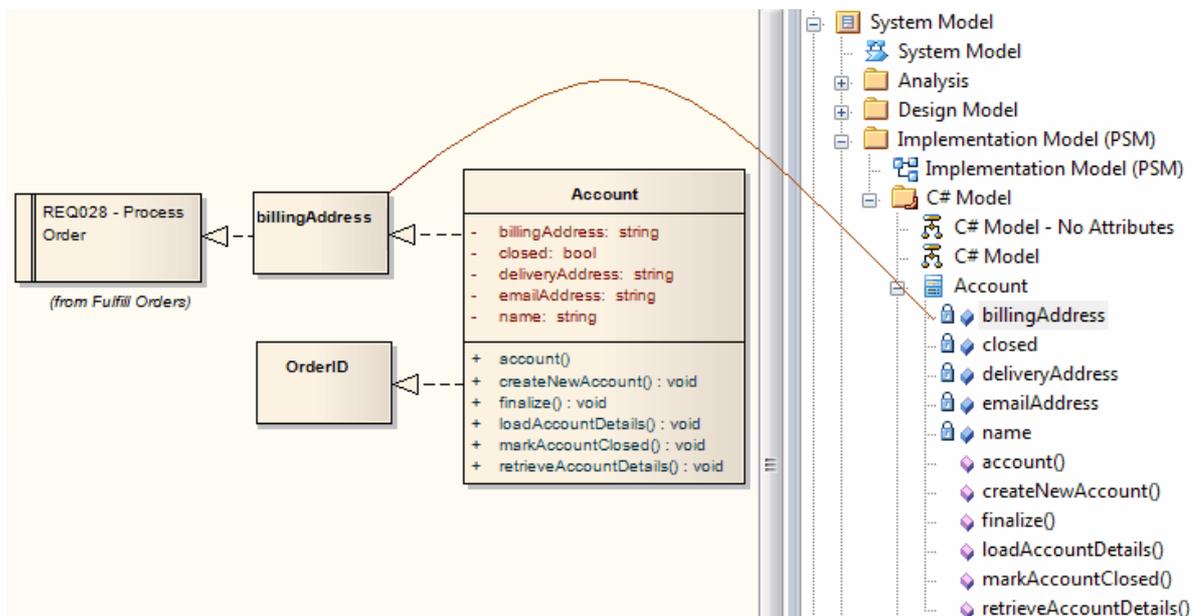
5.6.5.8 Create Object From Attribute

If you drag an attribute from the Project Browser onto an *Activity* diagram, the attribute generates an Object element of the same name. This is very useful for creating connectors between elements and specific attributes. For example, a Class element of stereotype *table* defines its fields as attributes; Requirement elements that define requirements for those fields can then be linked to the appropriate table fields via the attribute Object elements.

Topics:

Topic	Detail	See also
Explanation	<p>In the following diagram, the <i>billingAddress</i> Object was generated by dragging the <i>billingAddress</i> attribute from the <i>Account</i> Class in the Project Browser onto the diagram</p> <p>The user then created <i>Realize</i> relationships between the <i>Account</i> element and the <i>billingAddress</i> element, and between the <i>billingAddress</i> element and the <i>REQ028 Requirement</i> element</p>	

Example:



Notes:

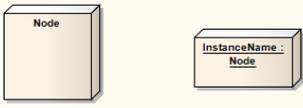
- From Enterprise Architect release 7.5, you can create this relationship more directly by linking an attribute in an element to another element, or linking two attributes in different elements

Learn More:

- [Connect to Element Feature](#) ⁷⁴²

5.6.6 Object Classifiers

Topics:

Image	Topic	Detail	See also
	Object Classifiers	<p>Many elements model classifications (such as Classes and Actors), whilst other elements model instances of such classifications (such as Objects, Actors again, and Sequence diagram objects). These instance elements represent real things in a run-time scenario; for example, a <i>Person</i> element named <i>Joe Smith</i>. In UML this is written as <i>Joe Smith: Person</i>.</p> <p>You can define a classifier first, and then instances of that classifier. Alternatively, as a model develops from a rough sketch to a detailed design, many objects become examples of a defined Class, so in the early analysis phase you might model a <i>Joe Smith</i> and a <i>Jane Smith</i>, and later a <i>Person Class</i> from which <i>Joe</i> and <i>Jane</i> are instantiated.</p> <p>Enterprise Architect enables you to associate an Object with its template element (its classifier), such as a Class. Doing this greatly increases the descriptive power of the model in capturing the functionality and responsibility of Objects at run-time and their associated state. For example, if you describe a <i>Person Class</i> with attributes such as <i>Age</i>, <i>Name</i>, <i>Address</i> and <i>Sex</i>, and functions such as <i>GetAge</i> and <i>GetName</i>, then when you associate your Object with the <i>Person Class</i> it is seen to have all the <i>Person Class</i> behavior and state (as well as inherited state and behavior from <i>Person's ancestors</i>).</p> <p>Elements that are classifiers and support instances of themselves at runtime can be dropped from the Project Browser as a link to the classifier itself, or a new instance of the classifier. The example below shows a linked Node element on the left and an instance of the Node on the right. Note that the instance is drawn like a simple element with the : <ElementName> displayed. If you name your instance it displays <InstanceName> : <ElementName>.</p>	

Notes:

- This is a powerful means of moving your model from the analysis phase into detailed design

5.6.6.1 Using Classifiers

If you right-click on an Object in a diagram, the element context menu displays the **Advanced | Instance Classifier** menu option. Select this option to choose a single element (generally a Class) as the classifier or template for this Object.

The **Select <Item> dialog** displays. Use this to set the instance classifier.

The Object name is then displayed as *Object: Classifier*, for example a Person object named Joe Smith is displayed as *Joe Smith: Person*.

Topics:

Topic	Detail	See also
If an object has a Classifier	<p>It is important to remember that an Object is only an instance of a classifier at runtime, so the appropriate attributes and operations are those of the classifier, not the Object. Therefore, in the context menu for the Object, if you select the Attributes or Operations menu options, the Attributes or Operations dialog displays for the classifier, not the Object.</p> <p>If you set the classifier for an Object in a Sequence diagram, when you add a message the drop-down list of available messages derived from the target Object come from the classifier, not the Object selected. This enables you to associate Sequence diagram objects with Classes and use the defined behavior of the Class to model actual behavior at run time.</p> <p>You can also select a message for a State Flow connector. The same rules apply as for Sequence diagram objects.</p> <p>Note that in the Message dialog you can also select to include messages defined in the target classifier's inheritance hierarchy.</p>	

Learn More:

- [Select <Item> Dialog](#)^[690]

5.6.7 Behavior

Enterprise Architect enables you to define an element's behavior through the element's operations and parameters. You can also define the behavior of more specific behavioral elements such as Activities, Interactions, Actions and Interaction Occurrences, through the **Behavior** and **Call** tabs of the element **Properties** dialogs.

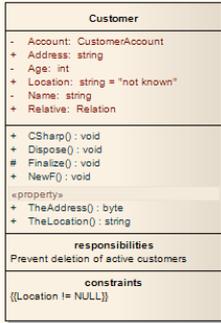
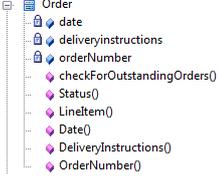
Learn More:

- [Operations](#)^[707]
- [Interactions and Activities](#)^[715]
- [Behavior Calls](#)^[715]
- [Behavior Parameters](#)^[717]
- [Behavior Call Arguments](#)^[716]

5.6.7.1 Operations

Operations are features of a Class or other element that represent the behavior or services an element supports. For a Customer Class, *UpdateCustomerName* and *GetCustomerAddress* can be operations. Operations have several important characteristics, such as type, scope (visibility), static, abstract and notes.

Topics:

Images	Topic	Detail	See also
	<p>How to Access Operations</p>	<p>If an element supports operations (typically Classes and Interfaces), the right-click context menu contains the Operations menu item. Select this to open the Operations dialog. Alternatively, press (F10) .</p>	<p>Operations dialog^[708]</p>
	<p>How Operations Appear in Diagrams</p>	<p>Elements with operations (typically Classes) display their features in diagrams in the manner shown below. Operations display in the second compartment of properties in colored text - the default color is dark green (for example, <i>Finalize() : void</i>). Some characteristics display in shorthand form; for example, static displays as \$, abstract as *.</p>	
	<p>Operations in the project Browser</p>	<p>In the Project Browser, Classes with operations have their features collected beneath them, each preceded by a pink box (). Right-click on an operation and select the Operation Properties context menu option to open the Operations dialog and edit details for the feature.</p> <p>From the Project Browser, you can drag operations onto new elements to give them the same operations.</p>	

Notes:

- If the parent element provides source or target roles for a connector, the connector can be attached to a specific operation. See the *Connect to Element Feature* topic

Learn More:

- [Behavior Parameters](#)^[717]
- [Connect to Element Feature](#)^[742]

5.6.7.1.1 Operations Dialog - General

The **Operations** dialog has seven tabs:

- General
- Parameters
- Behavior
- Advanced
- Tagged Values
- Pre-conditions and Post conditions (that is, Constraints)

The **General** tab of the **Operations** dialog enables you to define new operations and set the most common properties, including name, access type and return.

Field/Button	Usage	See also
Name	Display the selected operation name.	
Parameters	Display the parameter list. See Parameters Dialog for information regarding what this string can contain.	Parameters Dialog ^[717]
Edit Parameters	Open the Parameters dialog.	
Return Type	(Not shown for <i>State</i> or <i>State Machine</i> elements.) Display the data type returned by the operation. The type can be defined by the code language (data type) or by a classifier element. When you click on the drop-down arrow, the set of values in the list provides the appropriate data types. To select or define possible classifiers, either click on the Select Type option in the list, or click on the (...) (Select) button to display the Select <Item> dialog. To add new code language data types that can be displayed in this list, see the Data Types topic.	Instance classifier ^[690] Data Types ^[777]
(...) (Return Type Browse button)	Open the Select <Item> dialog to select the operation return type. (Not shown for <i>State</i> or <i>State Machine</i> elements.)	Instance classifier ^[690]
Action	Define the action of the operation: do , exit or entry . (For <i>State</i> or <i>State Machine</i> elements.)	State ^[917] State Machine ^[925]
Scope	Select Public/Protected/Private/Package .	
Stereotype	Specify an optional stereotype for this operation.	
Concurrency	Set the concurrency of the operation.	
Alias	Define an optional alias for the operation.	
Notes	Enter free text notes. You can format this text if necessary, using the Notes toolbar at the top of the field.	Notes toolbar ^[770]
Virtual/Abstract	If the operation's language is set to C++, map to the C++ <i>Virtual</i> keyword. Otherwise this option is <i>Abstract</i> , pertaining to an abstract function. (Not shown for <i>State</i> or <i>State Machine</i> elements.)	
Return Array	Indicate that the return value is an array.	

Field/Button	Usage	See also
	(Not shown for <i>State</i> or <i>State Machine</i> elements.)	
Synchronized	Specify a code engineering flag that relates to multi threading in Java. (Not shown for <i>State</i> or <i>State Machine</i> elements.)	
Static	Indicate that the operation is a static member. (Not shown for <i>State</i> or <i>State Machine</i> elements.)	
Const	Indicate that the return type of this method is constant. (Not shown for <i>State</i> or <i>State Machine</i> elements.)	
Pure	Indicate that C++ is pure virtual syntax - for example: <i>virtual void myFunction() = 0;</i> (Not shown for <i>State</i> or <i>State Machine</i> elements.)	
IsQuery	Indicate that this method does not modify the object. (Not shown for <i>State</i> or <i>State Machine</i> elements.)	
Operations	List the defined operations.	
Up/Down Buttons	Change the order of operations in the list.	
New	Create a new operation.	
Copy	Copy the currently selected operation.	
Save	Save a new operation, or save modified details for existing operation.	
Delete	Delete the currently selected operation.	

Notes:

- The **General** tab can vary according to the type of element you are adding an operation to. If defining operations for a data modeling table, see the Indexes, Triggers and Check Constraints topics
- If you make changes and do not save them, the **Cancel** button prompts you to confirm or cancel the changes, whilst the Close button closes the dialog immediately and does not save the changes
- If you are creating many operations, go to the **Attribute/Operations** page of the **Options** dialog (**Tools | Options | Source Code Engineering | Attribute/Operations**) and select the **After save, re-select edited item** checkbox. Now, when you create an operation and click on the Save button, the dialog fields clear ready for you to enter the details of the next operation. This helps you when you want to create operations quickly and might not necessarily want to fully define each one as you create it

Learn More:

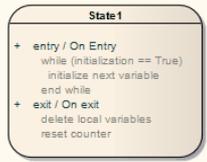
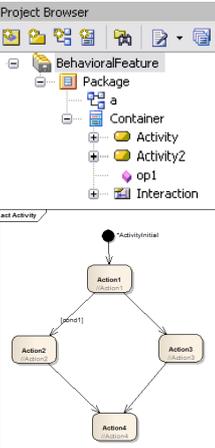
- [Operation Parameters](#) ^[717]
- [Behavior](#) ^[710]
- [Operation Constrains](#) ^[711]
- [Indexes](#) ^[1366]
- [Triggers](#) ^[1369]
- [Check Constraints](#) ^[1370]

5.6.7.1.1.1 Operations Dialog - Behavior

This topic illustrates how to elaborate a method's function in a diagram. The **Behavior** tab of the **Operations** dialog enables you to enter free text to describe the functionality of an operation. Use pseudo-code, structured English or just a brief description.

You can also use the **Behavior** tab to formally describe a method or State action and have the text appear under the method/action name in a diagram.

Topics:

Images	Topic	Detail	See also
	State Operations - Behaviour	A State's <i>do</i> , <i>entry</i> and <i>exit</i> operations optionally refer to other behaviors such as Activities, Interactions or Operations. Use the (...) (browse) button to invoke the Select <Item> dialog, and locate and select the required behavior.	Select <Item> Dialog 6901
	Associate with Behaviour	A Class operation can be associated with a behavior elsewhere in the model. Display the operation's Properties dialog, select the Behavior tab, click on the (...) (browse) button and select the required behavior from the Select Element dialog. The behavior is set in the Behavior field, and the operation is then set as the specification of the associated behavior. In behavioral code generation, the behavior of the associated behavioral element is generated as the <i>operation's</i> code. In the following illustration, <i>Op1</i> is associated with the <i>Activity Activity</i> .	
	Example Code	The generated code for Op1 is as follows: <pre>package Package; public class Container { public Container(){ } public void finalize() throws Throwable { } public void op1(){ /*Activity element(Activity1)'s behavior operation(op1)'s code*/ //Action1; if (cond1) { //Action2; } else { //Action3; } //Action4; } /*Activity element(Activity1)not rendered public void Activity2()</pre>	

Images	Topic	Detail	See also
		<pre> { // behavior is a Activity } public void Interaction() { // behavior is a Interaction } } //end Container </pre>	

Learn More:

- [Initial Code](#)^[71]

Topics:

Topic	Detail	See also
Usage	<p>On the Behavior tab of the Operations dialog, you can use the Initial Code field to enter code to be inserted into an operation body when the operation is first generated to file. After this point, forward code generation and synchronization do not replace the existing operation code with the Initial Code field.</p> <p>By default, the Initial Code field also is not imported into the model during reverse engineering, but you can select to import the field by selecting the Include method bodies in model when reverse engineering checkbox on the Options dialog.</p> <p>This field is most useful when combined with UML Patterns. Elements within a pattern often require the same stub code. Notice that the language specific patterns available from www.sparxsystems.com/resources/developers/uml_patterns.html include initial code for some of the defined operations. This helps speed up the process of applying patterns from model to implementation. The Initial Code section is also useful for ensuring that the generated code is directly compilable.</p>	<p>Options- Attributes/ Operations^[152]</p> <p>UML Patterns^[102]</p> <p>www.sparxsystems.com/resources/developers/uml_patterns.html</p>

5.6.7.1.1.2 Operations Dialog - Constraints**Topics:**

Topic	Detail	See also
Usage	<p>Operations can have pre- and post- conditions defined. For each type, give the condition a name, a type and enter notes.</p> <p>Constraints define the contractual behavior of an operation, what must be true before they are called and what is true after. In this respect they are related to the state model of a Class and can also relate to the guard conditions that apply to a transition.</p>	

5.6.7.1.2 Operation Tagged Values

Operations can have Tagged Values associated with them. Tagged Values offer a convenient extension mechanism for UML elements, so you can define any tags you like and then assign values to them using this form.

Tagged Values are written to the XML output, and can be input to other third party tools for code generation or other activities.

How To:

To add a Tagged Value for an operation, follow the steps below:

Step	Action	See Also
1	Click on the Tagged Values tab of the operation Properties dialog	
2	Click on the New Tags button The Tagged Value dialog displays	
3	In the Tag field, type the tag name (or select a defined tag from the drop-down list) In the Value field type the initial tag value	
4	Click on the OK button to confirm the operation	

Notes:

- Tagged Values are supported for attributes, operations, objects and connectors
- You can define custom tags using a predefined Tagged Value Type

Learn More:

- [Create a Custom Tagged Value](#)^[1113]

5.6.7.1.3 Override Parent Operations

In Enterprise Architect, you can automatically override methods from parent Classes and from realized Interfaces.

Access: [Element](#) | [Advanced](#) | [Overrides & Implementations](#)

Topics:

Topic	Detail	See also
Usage	In the Override Operations/Interfaces dialog, check the operations/interfaces to automatically override and click on the OK button. Enterprise Architect generates the equivalent function definitions in your child Class. You can configure Enterprise Architect to display this dialog each time you add a Generalization or Realization connector between Classes, and review their possible operations/interfaces to override/implement. Do this from the Links page of the Options dialog (select the Tools Options Links menu	

Topic	Detail	See also
	option).	

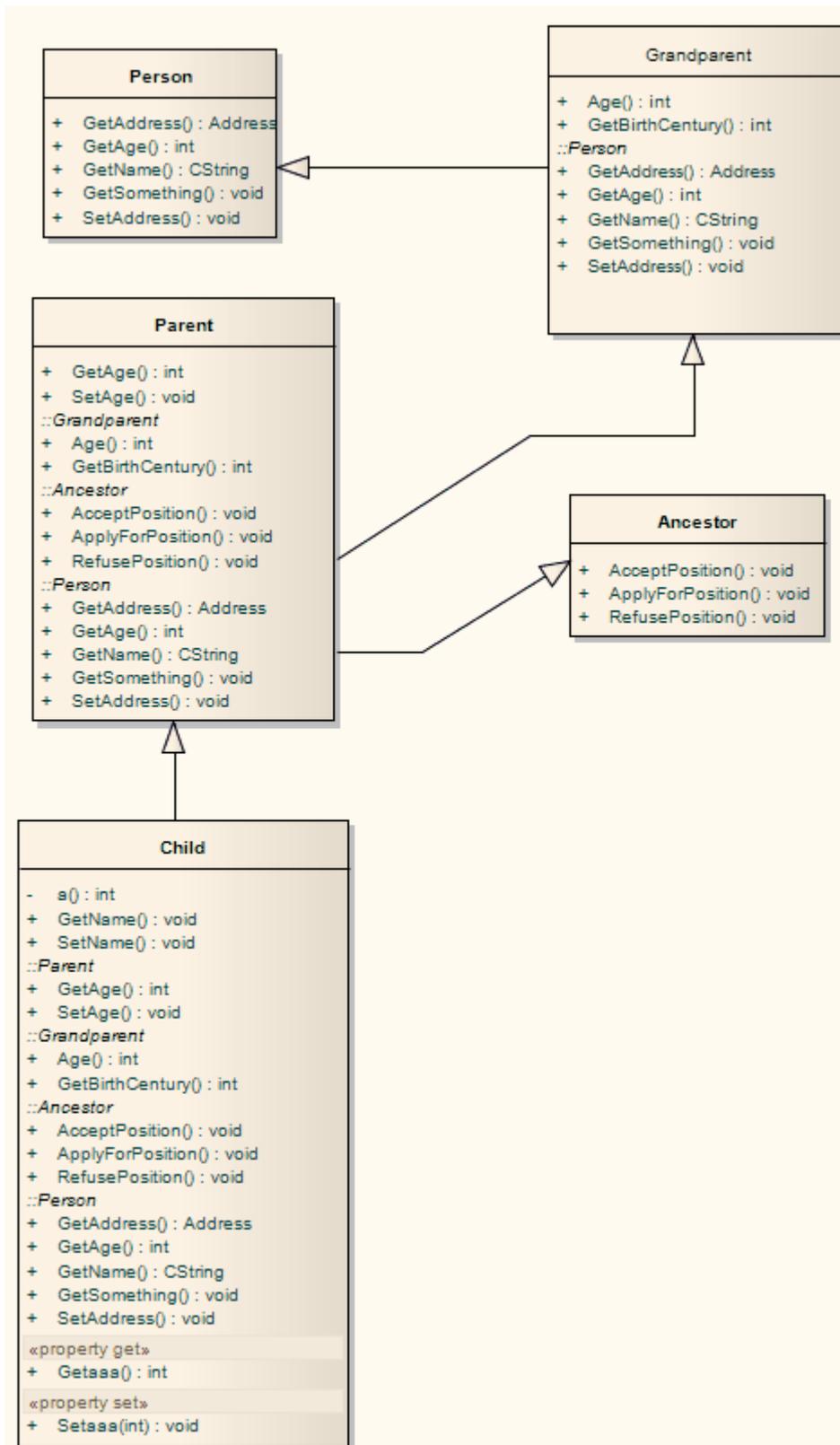
5.6.7.1.4 Display Inherited Operations

You can configure an element in a diagram to display the complete operation set obtained from all ancestors in the element's type hierarchy, as well as those directly owned.

Access: **Element | Feature Visibility | Shortcut: (Ctrl+Shift+Y)**

Example:

The following diagram illustrates this behavior when enabled for each element in a simple hierarchy.



5.6.7.2 Interactions and Activities

The behavioral aspects of Interactions and Activities are modeled using the **Behavior** tab of the element **Properties** dialog, which enables you to assign parameters and return types to the elements.

Topics:

Topic	Detail	See also
Usage	<p>Use the Edit Parameters button to edit an element's parameters. When you create a new parameter using the dialog, it internally creates an Activity Parameter Node for an Activity or an Interaction Parameter for an Interaction. In the Return field, click on the drop-down arrow and select the return type of the element.</p> <p>The specification field is populated automatically when an operation is associated with the activity as a behavior.</p>	<p>Parameters Dialog^[717]</p> <p>Activity Parameter Nodes^[875]</p> <p>Operations Dialog - Behaviour^[710]</p>

5.6.7.3 Behavior Calls

A behavior call is the invocation of a behavior. You can represent an invocation with a Call Operation Action (Operation), Call Behavior Action (Activity) or Interaction Occurrence (Interaction) element. You model the properties of the behavior call using the **Call** tab of the element **Properties** dialog.

Use To:

- Edit Arguments
- Re-associate the call with a different behavior
- Synchronize the arguments with the parameters in the associated behavior

Topics:

Topic	Detail	See also
Usage	<p>Click on the Edit button to create and delete arguments, and relate them to a corresponding parameter in the associated behavior.</p> <p>Click on the (...) (Select Behavior) button to re-associate the invocation with a different behavior or to remove any association with the current behavior.</p> <p>The Interaction Occurrence panel is displayed only for Interaction Occurrence elements. It enables you to enter the return value and attribute of the behavior call.</p>	

Learn More:

- [Action Notation](#)^[866]
- [Interaction Occurrence](#)^[906]
- [Arguments](#)^[716]
- [Associate with Different Behavior](#)^[716]
- [Synchronize Arguments](#)^[716]

5.6.7.3.1 Associate with Different Behaviors

On the **Call** tab of the **Behavior Call Properties** dialog, when you click on the (...) (**Select Behavior**) button the **Select <Item>** dialog displays, listing all available behaviors in the model.

Topics:

Topic	Detail	See also
Usage	Select <none> to remove any existing association between an invocation and a behavior, or select another classifier to re-associate the invocation with a different behavior. The Synchronize with Parameters button is enabled only if a valid behavior is identified in the Behavior field.	

Learn More:

- [Select <Item> Dialog](#)^[690]

5.6.7.3.2 Synchronize Arguments

Topics:

Topic	Detail	See also
Usage	On the Call tab of the element Properties dialog, click on the Synchronize with Parameters button to synchronize the number of arguments in the invocation element with the number of parameters in the associated behavior. This automatically creates or deletes arguments based on the number of parameters in the behavior. If any arguments are to be deleted, Enterprise Architect prompts you to confirm the operation. Click on the Yes button to confirm.	

Notes:

- The **Synchronize with Parameters** button is enabled only if the invocation is associated with a valid behavior, as identified in the **Behavior** field

5.6.7.4 Behavior Call Arguments

You define the arguments of a Behavior Call using the Arguments dialog, which you display by clicking on the **Edit** button on the Call tab of the element Properties dialog.

How To:

To define the arguments of a Behavior Call, follow the steps below:

Step	Action	See Also
1	In the Name field, type the name of an argument to map to the behavior	
2	In the Parameters field, click on the drop-down arrow and select a behaviors parameter from the list of parameters associated with the behavior	

Step	Action	See Also
3	In the Value field, set any required value.	
4	If a diagram is displayed, and if required, select the Show in current diagram checkbox to add an Action Pin on the diagram	Action Pin 869
5	Click on the Save button	
6	If appropriate, click on the New button and repeat steps 1 to 5 for another argument: parameter mapping	

Notes:

- If you attempt to map a newly created argument to a parameter that is already associated with a different argument, Enterprise Architect identifies the mapping and prompts you to confirm that you intend to change the association

Learn More:

- [Behavior Calls](#) 715

5.6.7.5 Behavior Parameters

This topic area describes the facilities for defining, editing and extending the parameters of behavioral operations, Activities and Interactions.

Learn More:

- [Parameters Dialog](#) 717
- [Parameter Tagged Values](#) 718
- [Operation Parameters By Reference](#) 719

5.6.7.5.1 Parameters Dialog

The **Parameters** dialog enables you to define the parameters of an operation, Activity or Interaction. The parameter list is reproduced in code in the order the parameters appear in the parameters list, so use the **Up** and **Down** buttons to move parameters into their required positions. Additionally, you can select the **Add new to end** checkbox to force new parameters to appear at the end of the list instead of the top.

Reference:

Field	Usage	See also
Name	Indicates the parameter name.	
Type	Select the data type of the parameter. Alternatively, click on the (...) button and select the element classifier to define the type.	
Default	Indicates an optional default value for the parameter.	
Stereotype	Indicates a stereotype name, specification of a Stereotype name can also be chosen via the drop-down list.	
Kind	Indicate the way a parameter is passed to a function:	

Field	Usage	See also
	<ul style="list-style-type: none"> • In = By Value • InOut = By Reference • Out is passed by Reference, but only the return value is significant. 	
Fixed	Set the parameter to <i>const</i> , even if passed by reference.	
Alias	Indicates an optional alias for the parameter.	
Add new to end	Place new parameters at the end of the list instead of the start.	
Multiplicity	Display the Multiplicity dialog, to specify the multiplicity of the parameters.	
Notes	Indicates any additional notes on the parameter.	

Below is information regarding the Multiplicity dialog.

Reference:

Field	Usage	See also
Lower bound	Define a lower limit to the number of elements allowed in the collection.	
Upper bound	Define an upper limit to the number of elements allowed in the collection.	
Allow Duplicates	Indicate that duplicates are allowed. Maps to the UML property <code>isUnique</code> , value <code>FALSE</code>).	
Multiplicity is Ordered	Indicate that the collection is ordered.	

Notes:

- Set the amount of parameter detail to display in a specific diagram using the **Show Parameter Detail** drop-down list on the **Diagram Properties** dialog. The setting applies only to the current diagram. The default is to show the type only

Learn More:

- [Parameter Tagged Values](#)^[718]
- [Operation Parameters by Reference](#)^[719]
- [Visible Class Members](#)^[575]

5.6.7.5.1.1 Parameter Tagged Values

Behavioral parameters can have Tagged Values associated with them. **Tagged Values** offer a convenient extension mechanism for UML elements; you can define any tags you like and then assign values to them using this form.

Tagged Values are written to the XML output, and can be input to other third party tools for code generation or other activities.

How To:

To add a Tagged Value for a parameter, follow the steps below:

Step	Action	See Also
1	Double-click on the operation, Activity or Interaction containing the parameter in a diagram or in the Project Browser The Properties dialog displays	
2	Click on the Tagged Values tab, which shows the Tagged Values for the selected object and its parameters	
3	Click on the required parameter in the Parameters compartment of the Tagged Values tab, and click on the New Tags button The Tagged Value dialog displays	
4	In the Tag field, type the tag name (or select a defined tag from the drop-down list) In the Value field type the initial tag value	
5	Click on the OK button to confirm the Tagged Value	

Notes:

- Tagged Values are supported for attributes, operations, objects and connectors
- Custom tags can be created using a predefined Tagged Value Type

Learn More:

- [Create a Custom Tagged Value](#)^[1113]

5.6.7.5.2 Operation Parameters by Reference

Access: **Tools | Options | Objects : Feature reference indicator**

Topics:

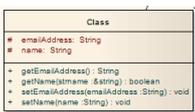
Image	Topic	Detail	See also
	Usage	<p>You can select to highlight parameters declared as 'Kind: <i>inout</i>' or 'Kind: <i>out</i>' with an additional user-defined prefix or suffix.</p> <p>If you select the Highlight References checkbox, you can also indicate whether a prefix or suffix should be used, and the actual reference character to use. In the example above, the & character has been set as a prefix.</p> <p>When you declare a parameter of type <i>inout</i>, it is assumed you are passing the parameter by reference rather than by value. If you have elected to highlight references, then this is displayed in the Diagram View.</p>	Objects ^[432]

Image	Topic	Detail	See also
		The example to the left shows that, in the <i>getName</i> operation, the parameter <i>strName</i> is a string reference, and is highlighted using the chosen character and position.	

Notes:

- This facility currently applies to operations only

5.6.8 In-place Editing Options

This topic explores the tasks that can be performed using in-place editing of elements on a diagram in Enterprise Architect. The tasks include:

- [View Properties](#) ^[721]
- [Edit Element Item Name](#) ^[722]
- [Edit Feature Stereotype](#) ^[722]
- [Edit Feature Scope](#) ^[723]
- [Edit Attribute Keyword](#) ^[724]
- [Edit Operation Parameter Keyword](#) ^[724]
- [Insert Operation Parameter](#) ^[726]
- [Edit Parameter Kind](#) ^[725]
- [Insert New Feature](#) ^[726] (Attribute or Operation)
- [Insert Maintenance Feature](#) ^[727]
- [Insert Testing Features](#) ^[728]
- [Delete Selected from Model](#) ^[721]

5.6.8.1 In-place Editing Tasks

How To:

To use the in-place editing menu options, follow the steps below:

Images	Step	Action	See Also
	1	Open the diagram containing the element	
	2	Click on the element, and on the item to manipulate within the element The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Edit and manipulate the items in the element, either by pressing the appropriate keyboard keys or by right-clicking on the highlighted item and choosing a task from the Element Items context menu The available commands are listed in the table below	

Reference:

Menu option...	Usage	Shortcut	See also
Edit Selected	Change the name, scope or stereotype of the element or element item	F2	
View Properties	Display the dialog containing details of the element	Enter	
Insert New After Selected	Insert a new item in the element	Insert	
Find in Project Browser	Locate the item in the Project Browser		
Add Attribute	Add an attribute to the element	Ctrl + Shift + F9	
Add Operation	Add an operation to the element	Ctrl + Shift + F10	
Add Other	Insert a feature on the specific element item, such as Maintenance features and Testing features	Ctrl + F11	
Delete Selected from Model	Delete the selected item from the model	Delete	
View Source code	Display the source code for the element, in the default external editor	F12	The Source Code Viewer ^[1408]
Set Breakpoint	Set a breakpoint on a highlighted operation (including a breakpoint, start recording marker, end recording marker or stack auto-capture marker)		Breakpoint and Marker Management ^[1436] Marker types ^[1642]
	Navigate Diagram Selection, to navigate the diagram between elements without having to use the mouse	Ctrl + Shift + arrow key	
	Toggle element highlight option on and off	Shift + Enter	

Other options that are available while editing element attributes or operations in a diagram include:

Menu options	Usage	Shortcut	See also
	Accept current changes	Enter	
	Accept current changes and open a new slot to add a new item	Ctrl + Enter	
	Abort the edit, without save	Esc	
	Move the cursor between segments of a feature definition	Tab or Shift + Tab	
	Display the context menu for in-place editing	Shift + F10	
	Invoke the Select <Item> dialog	Ctrl + Shift + Space	Select <Item> Dialog ^[690]

Menu options	Usage	Shortcut	See also
	Display an autocompletion list for a feature classifier	Ctrl + Space	

Notes:

- Most of the in-place editing menu commands have keyboard alternatives; for many of them, if the selected item happens to be off-screen when you press the appropriate keys, the diagram automatically scrolls to show the whole element, so that you can see what you are changing

5.6.8.2 Edit Element Item Name**How To:**

To change the name of the element, or the name of an operation or attribute, directly from the diagram, follow the steps below

Images	Step	Action	See Also
	1	Open the diagram containing the element	
	2	Click on the element and on the name to change within the element The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Right-click on the item The context menu displays	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the item directly from the diagram The name of the attribute or operation is highlighted	
	5	Delete or type over the name	
	6	Press (Enter) to accept the change, or (Esc) to cancel the change	

5.6.8.3 Edit Feature Stereotype**How To:**

To change the stereotype of an operation or attribute directly from the diagram, follow the steps below:

Images	Step	Action	See Also
	1	Open the diagram containing the element	
	2	Click on the element, and on the item to edit within the element The item line is highlighted in a lighter shade (the default is white), to	

Images	Step	Action	See Also
		indicate that it has been selected	
	3	Right-click on the item The context menu displays	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the attribute or operation directly from the diagram The name of the item is highlighted	
	5	Move the cursor to the position before the name or within the existing attribute or operation stereotype (denoted by «stereotype name»)	
	6	Delete or type over the previous name to change the stereotype name of the attribute or operation	
	7	Press (Enter) to accept the change or (Esc) to cancel the change	

Notes:

- You can assign multiple stereotypes by including a comma-separated list inside the stereotype markers

5.6.8.4 Edit Feature Scope

To rapidly change the scope of an attribute or operation directly from the diagram, follow the steps below

How To:

Images	Step	Action	See Also
	1	Open the diagram containing the element	
	2	Click on the element and on the item to edit within the element The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Right-click on the item The context menu displays	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the attribute or operation directly from the diagram The name of the item is highlighted.	
	5	Move the cursor to the scope of the item and delete the previous entry	
	6	Reassign the entry by typing in one of the following symbols: <ul style="list-style-type: none"> • + indicates that the scope is Public • - indicates that the scope is Private • ~ indicates that the scope is Package • # indicates that the scope is Protected 	
	7	Press (Enter) to save the change, or (Esc) to cancel the change	

Images	Step	Action	See Also
		The diagram is updated to reflect the changes (also see the <i>catalogNumber</i> attribute in the screen illustrations to the left)	

5.6.8.5 Edit Attribute Keyword

You can add features such as attribute keywords and classifiers directly to an element, using the **Element Keywords and Classifiers** menu. This enables you to rapidly assign details element item by element item, directly from a diagram.

How To:

To add features directly to an element, follow the steps below:

Images	Step	Action	See Also
	1	In Enterprise Architect, open the diagram containing the element	
	2	Click on the element, and on the attribute to edit within the element The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Right-click on the item The context menu displays.	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the attribute directly from the diagram The name of the attribute is highlighted	
	5	Right-click on the attribute name to display the context menu	
	6	From the context menu, you can: <ul style="list-style-type: none"> Change the attribute classifier to static or fixed - select the static or fixed menu options as appropriate; the diagram is updated to reflect the changes Display the Class properties - click on the Goto Definition menu option; Enterprise Architect locates the Class in the Project Browser and opens its Properties dialog <p>If the data type is a raw data type, Enterprise Architect displays the message: <i>The data type is a raw data type</i></p>	Properties Dialog ^[660]

5.6.8.6 Edit Operation Parameter Keyword

You can directly edit operation classifiers by element, using the in-place editing menu. This enables you to rapidly assign parameter keywords.

How To:

To directly edit operation classifiers by element, follow the steps below:

Images	Step	Action	See Also
	1	Open the diagram containing the element	
	2	Click on the element, and on the operation to edit within the element The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Right-click on the item The context menu displays	
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the operation directly from the diagram The name of the operation is highlighted	
	5	Right-click on the data type of a parameter to display the context menu	
	6	From the context menu you can: <ul style="list-style-type: none"> • Change the operation classifier by clicking on the appropriate menu option - static, isquery, abstract or fixed; the diagram is updated to reflect the changes • Display the Class properties - click on the Goto Definition menu option <p>If the data type is Class, Enterprise Architect locates the Class in the Project Browser and opens its Properties dialog</p> <p>If the data type is a raw data type, Enterprise Architect displays the message <i>This data type is a raw data type</i></p> <p>If the data type is not defined in the model, the message is: <i>The data type is not defined in the model</i></p>	Properties Dialog ^[660]

5.6.8.7 Edit Parameter Kind

You can edit operation parameter kinds such as (*in*), (*inout*) and (*out*) directly from a diagram element by element, using the **Element Keywords and Classifiers** menu. This enables you to rapidly assign the parameter directly from a diagram.

How To:

To edit operation parameter kinds directly from a diagram, follow the steps below:

Images	Step	Action	See Also
	1	In Enterprise Architect, open the diagram containing the element	
	2	Click on the element, and on the operation to edit within the element The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Right-click on the item The context menu displays	

Images	Step	Action	See Also
	4	Select the Edit Selected menu option (or press (F2)) to enable you to edit the item directly from the diagram The name of the item is highlighted	
	5	Right-click on the item name to display the context menu	
	6	Select the appropriate menu option for the parameter kind value: (in), (inout) and (out) The diagram is updated to reflect the change	

5.6.8.8 Insert New Feature

How To:

To add attributes and operations to a Class diagram element, follow the steps below:

Images	Step	Action	See Also
	1	Open the diagram containing the element to which you are adding an attribute or operation	
	2	Click on the element and, within the element, on the item after which to insert the operation or attribute The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Either: <ul style="list-style-type: none"> • Press (Insert) or • Right-click on the selected element item to display the context menu and select the Insert New After Selected menu option Enterprise Architect inserts a new data line in the diagram, underneath the selected item	
	4	Type in the relevant information for the attribute or operation Use the (Tab) and (Tab+Shift) keys to move the cursor through the segments of the attribute or operation, and select the classifier from either the: <ul style="list-style-type: none"> • Autocompletion list (press the (Ctrl+Space) keys) or • Select <Item> dialog (press the (Ctrl+Shift+Space) keys) 	Autocompletion List^[1395] Select <Item> Dialog^[690]
	5	Press (Enter) to accept the change or (Esc) to cancel the change The diagram is updated to reflect the changes	

5.6.8.9 Insert Operation Parameter

You can add operation parameters to an operation through the in-place editing options, using hotkey commands or menu shortcuts.

How To:

To add parameters to operations in a Class diagram element, follow the steps below:

Images	Step	Action	See Also
	1	Open the diagram containing the element	
	2	Click on the element, and on the operation to update within the element The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Press (F2), or right-click on the selected item to display the context menu and select the Edit Selected option	
	4	Move the cursor inside the parameter brackets and type the parameter name, followed by a colon (for example, bks: for a vector containing books)	
	5	Give the parameter a type Place the cursor after the colon at the end of the name and add the classifier; either: <ul style="list-style-type: none"> • Press (Ctrl+Space) to invoke the classifier autocompletion list • Press (Ctrl+Shift+Space) to select a classifier from the Select <Item> dialog or • Right-click the mouse to display the inline editing options context menu and select the Insert Classifier option The Select <Item> dialog displays	Autocompletion List ^[1399] Select <Item> Dialog ^[690]
	6	Press (Enter) to accept the change or (Esc) to cancel the change The diagram is updated to reflect the changes.	

5.6.8.10 Insert Maintenance Feature

How To:

To rapidly assign maintenance details such as Defects, Changes, Issues and Tasks directly to an element from a diagram, follow the steps below:

Images	Step	Action	See Also
	1	Open the diagram containing the element	
	2	Click on the element name The name is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Either: <ul style="list-style-type: none"> • Press (Ctrl+F11) or • Right-click on the highlighted name to display the context menu, and select the Add Other option The Insert Feature dialog displays	
	4	Click on the appropriate radio button option to associate the required maintenance feature with the element item	
	5	Click on the OK button	

Images	Step	Action	See Also
		The <Maintenance Feature> details for <element> dialog displays	
	6	Complete the fields to define the maintenance activity, and then click on the Apply button To create a subsequent maintenance activity of this type, click on the New button	
	7	When you have defined all of the maintenance activities of this type, click on the OK button The maintenance details are added to the element	

Notes:

- To ensure that the maintenance items are visible in the diagram element, as shown in step 7, select the **Maintenance** checkbox on the Elements tab of the Diagram Properties dialog

Learn More:

- [Show Maintenance Script in Diagram](#)^[1717]

5.6.8.11 Insert Testing Features**How To:**

To rapidly add testing features such as Unit, Integration, System, Acceptance and Scenario tests to an element directly from a diagram, follow the steps below:

Images	Step	Action	See Also
	1	Open the diagram containing the element	
	2	Click on the element The element name is highlighted in a lighter shade (the default is white), to indicate that it has been selected	
	3	Either: <ul style="list-style-type: none"> Press (Ctrl+F11) or Right-click on the highlighted name to display the context menu and select the Add Other option The Insert Feature dialog displays	
	4	Click on the appropriate radio button option to associate the required testing feature with the element	
	5	Click on the OK button The Testing window opens, showing the appropriate panel for the type of test selected	
	6	Complete the fields to define the test activity, and then click on the Save icon in the window toolbar The test is added to the element	The Testing Workspace ^[1695]

Images	Step	Action	See Also
	7	To create a subsequent test activity of this type, click on the New icon To add items for other types of test, click on the appropriate tab	

Notes:

- To ensure that the test items are visible in the diagram element, as shown in step 7, select the **Testing** checkbox on the Elements tab of the Diagram Properties dialog

Learn More:

- [Show Test Script Compartments](#)^[1707]

5.6.9 Linked Documents

In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, you can link an RTF document to any UML element in the model.

All editions of Enterprise Architect provide an additional UML Artifact - Document Artifact - that can contain an RTF document internally.

You create linked documents from Linked Document Templates, which you define with the Document Template Editor.

Topics:

Topic	Detail	See also
Usage	<p>Documents created via the Document Artifact element are rendered into RTF Documentation by selecting the Linked Document checkbox in the RTF Style Template Editor.</p> <p>The Linked Document checkbox is within the Element hierarchy, towards the end. Remember that checkboxes can be moved up and down the hierarchy to position information in the generated document as you require. In some templates, the Linked Document checkbox is only available as a child of the External Requirements checkbox.</p> <p>The linked document is rendered into the RTF documentation at:</p> <p><i>linked document ></i></p> <p><i><linked document</i></p>	

Notes:

- When you have saved the document, an **A** symbol displays in the bottom right corner of the element

Learn More:

- [Document Artifact](#)^[951]
- [Create Linked Document Templates](#)^[734]
- [Edit Linked Document Templates](#)^[734]

- [Select Components For Documentation](#)^[1737]
- [Create Document Artifact](#)^[730]
- [Link Document to UML Element](#)^[730]
- [Edit Linked Documents](#)^[731]
- [Hyperlink From Linked Document](#)^[732]
- [Create Element From Document](#)^[732]
- [Replace or Delete Documents](#)^[733]
- [Generate RTF Documentation Dialog](#)^[1732]

5.6.9.1 Create Document Artifact

Topics:

Topic	Detail	See also
Usage	<p>You create a Document Artifact element In a Component or Deployment diagram.</p> <p>Drag and drop the Document Artifact element from the Toolbox into your diagram.</p> <p>Double-click on the Document Artifact element. The Linked Document Editor opens, with the New Linked Document dialog.</p> <p>In the Copy template field, click on the drop-down arrow and select a previously-created Linked Document Template. Click on the OK button.</p>	<p>Component Diagram^[807]</p> <p>Deployment Diagram^[804]</p> <p>Linked Document^[731]</p> <p>Create Linked Document Templates^[734]</p> <p>Edit Linked Document Templates^[734]</p>

5.6.9.2 Link Document to UML Element

How To:

Step	Action	See Also
1	<p>Click on an element in the Project Browser, Diagram List, Package Browser, Model Search or diagram, and either:</p> <ul style="list-style-type: none"> • Select the Element Linked Document menu option • In the Notes window, click on the Linked Document icon in the toolbar • Press (Ctrl+Alt+D) or • Right-click and select the (Create) Linked Document option from the context menu <p>Alternatively, open the element Properties dialog and click on the Linked Document icon in the toolbar in the Notes field</p>	
2	The New Linked Document dialog displays	
3	Select the previously-created template from which to create the document	
4	<p>Click on the OK button</p> <p>The Linked Document editor displays</p>	Linked Document Editor ^[731]
5	Enter the text of the document	

Notes:

- This operation is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- When you have saved the document, an **A** symbol displays in the bottom right corner of the element

Learn More:

- [Create Linked Document Templates](#) ^[734]
- [Edit Linked Document Templates](#) ^[734]

5.6.9.3 Edit Linked Documents

Topics:

Topic	Detail	See also
Usage	<p>Enterprise Architect provides a Windows-like word processor to help you edit Linked Documents. This is a simplified version of the RTF Style Template Editor, and it provides the same convenient features.</p> <p>The main difference between the two editors is that you access the RTF Style Template Editor features through a menu bar at the top of the screen, whilst you access the Linked Document Editor features through a context menu. To access the context menu, just right-click anywhere on the document.</p> <p>You can format and edit the document in a number of ways, and add links from and references to the content of the document. For example, you can highlight a word or term in the linked document and select Create Glossary Definition to create a glossary definition for that term. Anyone reading the document can, if they check for the term in the Project Glossary, read the definition.</p>	

Learn More:

- [Create a Glossary Definition](#) ^[362]
- [Glossary Dialog](#) ^[363]
- [Scroll Through Text](#) ^[1744]
- [File and Print Options](#) ^[1745]
- [Cut and Paste Options](#) ^[1746]
- [Image and Object Imports](#) ^[1749]
- [Character Formatting](#) ^[1750]
- [Paragraph Formatting](#) ^[1751]
- [Tab Support](#) ^[1753]
- [Page Breaks and Repagination](#) ^[1754]
- [Insert Headers and Footers](#) ^[1754]
- [Insert Bookmarks](#) ^[1755]
- [Table Commands](#) ^[1756]
- [Sections and Columns](#) ^[1759]
- [Stylesheets and Table of Contents](#) ^[1759]
- [Text/Picture Frame and Drawing Objects](#) ^[1763]
- [Search/Replace Commands](#) ^[1764]
- [Hyperlink From Linked Document](#) ^[732]
- [Create Elements From Linked Documents](#) ^[732]

5.6.9.4 Hyperlink From Linked Document

Topics:

Topic	Detail	See also
How To	<p>To do this, click on the object in the Project Browser and drag it to the point at which to create the hyperlink. The linked document editor automatically creates the hyperlink, using the object name as the hyperlink text. You can edit this text if required.</p> <p>Similarly, you can create a hyperlink to an element in the model by highlighting the link text in the linked document, right-clicking on it and selecting the Create Link to Existing Element context menu option. This displays the Select Classifier dialog, from which you select the element to link to.</p> <p>In either case, when you next open the document, you can double-click on the hyperlink to locate and highlight the object in the Project Browser. You can then perform all normal operations on the object, including opening any linked document on the highlighted element.</p> <p>You can also create a hyperlink to a wide range of additional objects, such as web pages, Help files, Model Searches and Team Review Forums, by highlighting the appropriate text and then selecting the Create New Hyperlink context menu option. This displays the Hyperlink Details dialog. When specifying the object location, you can use full paths or local (path substitution) paths.</p> <p>For an alternative method of creating a hyperlink to an external document, Help file or web page, see the Hyperlinks and Bookmarks topic.</p>	<p>Select <Item> Dialog^[690]</p> <p>Model Searches^[482]</p> <p>Team Review^[216]</p> <p>Hyperlinks^[1292]</p> <p>Hyperlinks and Bookmarks^[1755]</p>

5.6.9.5 Create Element From Document

Using the **Linked Document Editor**, you can create document-specific elements and diagrams in the **Project Browser**, with hyperlinks from the document to the created item. When you click on the hyperlink, the element or diagram is highlighted in the **Project Browser**. The element or diagram is created in the same package as the element for which the linked document was created.

Topics:

Topic	Detail	See also
Usage	<p>You can create and link to any type of element or diagram, but the facility has specific options for the following element types:</p> <ul style="list-style-type: none"> • Class • Requirement • Issue <p>You can create the same arrangement with existing elements, diagrams and packages by dragging them from the Project Browser into the text of the document, creating a hyperlink with the item name as the text</p>	<p>Class^[941]</p> <p>Requirement^[1154]</p> <p>Issue^[1719]</p> <p>Hyperlink From Linked Document^[732]</p>

How To:

Step	Action	See Also
1	Open the linked document, either from a Document Artifact element or through the context menu for an existing element (Corporate, Business and Software Engineering, System Engineering and Ultimate editions)	Document Artifacts ^[730] Linking Document to UML Element ^[730]
2	Enter some text, including appropriate text to act as the link (such as the element or diagram name)	
3	Highlight the appropriate text and right-click on it The editor context menu displays	
4	Select the Create New menu option, and the required submenu option If you select the: <ul style="list-style-type: none"> • Class, Requirement or Issue option, the corresponding element is immediately created in the Project Browser • Other option, the New Element dialog displays; specify the element type and - if appropriate - stereotype, and click on the Create button • Diagram option, the New Diagram dialog displays; specify the diagram type and click on the OK button 	New Elements ^[627] New Diagrams ^[568]
5	The highlighted text is now a hyperlink Click on the link to highlight the new element or diagram in the Project Browser You can now edit or expand the element or diagram as required	

5.6.9.6 Replace or Delete Documents

If a linked document is out of date, you can either edit the text or replace the entire contents from another file.

How To:

To replace the contents, follow the steps below:

Step	Action	See Also
1	Click in the body of the document and press (Ctrl+A) to select all the document text	
2	Press (Delete)	
3	Right-click and select the File Import context menu option The Windows Open dialog displays, in which you can browse for the file to import into the document	
4	Click on the Save icon in the Linked Document screen toolbar	

To delete the linked document, follow the steps below:

Step	Action	See Also
1	Click on an element in the Project Browser or diagram, and either: <ul style="list-style-type: none"> select the Element Delete Linked Document menu option or right-click and select the Delete Linked Document context menu option 	
2	Enterprise Architect prompts you to confirm the deletion; click on the Yes button. If required, you can now create another linked document for the element.	

Learn More:

- [Edit Linked Documents](#)^[731]

5.6.9.7 Create Linked Document Templates

Topics:

Topic	Detail	See also
Usage	<p>Under the <i>Templates</i> folder, right-click on the Linked Document Templates icon and click on the Create Template context menu option.</p> <p>Enter a name for your template, or select a previously-created template. Click on the OK button.</p> <p>You can group your templates into folders. Right-click on your newly created template and select the Assign Template to Group context menu option. Enter a category name and click on the OK button.</p> <p>You can also modify and delete the templates using the context menu options.</p>	Edit Linked Document Templates ^[734]

Notes:

- You can transport these linked document templates between models, using the Export Reference Data and Import Reference Data options

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.6.9.8 Edit Linked Document Templates

Double-click on a previously created template in the Resource View to invoke the Linked Document Template Editor.

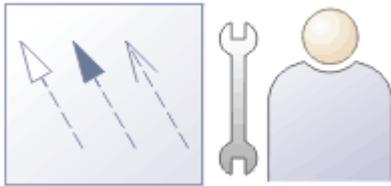
The Document Template Editor is built into Enterprise Architect.

Learn More:

- [Scroll Through Text](#)^[1744]
- [File and Print Options](#)^[1745]
- [Cut and Paste Options](#)^[1746]
- [Image and Object Inserts](#)^[1749]
- [Character Formatting](#)^[1750]

- [Paragraph Formatting](#)^[1751]
- [Tab Support](#)^[1753]
- [Page Breaks and Repagination](#)^[1754]
- [Headers and Footers](#)^[1754]
- [Hyperlinks and Bookmarks](#)^[1755]
- [Table Commands](#)^[1756]
- [Sections and Columns](#)^[1759]
- [Stylesheets and Table of Contents](#)^[1759]
- [Text/Picture Frame and Drawing Objects](#)^[1763]
- [Search/Replace Commands](#)^[1764]

5.7 Connectors



UML connectors, along with elements, form the basis of a UML model. Connectors link elements together to denote some kind of logical or functional relationship between them. Each connector has its own purpose, meaning and notation and is used in specific kinds of UML diagrams.

Topics:

Topic	Detail	See also
Off-Page Connector	<p>UML, and therefore Enterprise Architect, does not have a connector that continues activity flow between two diagrams</p> <p>In creating a model diagram, if it becomes necessary to continue flow to another diagram, you should consider revising and simplifying the structure of the process so that groups of Actions are captured in composite Activity elements, and each group of Actions is modeled within the child diagram of an Activity</p> <p>BPMN, however, does enable you to create off-page connectors</p> <p>You can also use the Suppress Line Segments menu option to indicate continuation of flow in a large diagram that, when printed, occupies several pages; be aware that these options are purely diagrammatic and do not indicate any diagram relationships in any of the relationship tools</p>	<p>BPMN ^[1228]</p> <p>Style Menu Section ^[739]</p>

Learn More:

- [Connector Context Menu](#) ^[736]
- [Connector Tasks](#) ^[740]
- [Connector Properties](#) ^[756]

5.7.1 Connector Context Menu

If you right-click on a connector in a diagram, the connector context menu displays. This provides quick access to some important functions.

Topics:

Topic	Detail	See also
Context Menu	<p>The menu is split into up to seven distinct sections:</p> <ul style="list-style-type: none"> • Add-Ins - displays in the first section only if you have Add-Ins installed and registered, such as Eclipse • Properties • Type Specific • Advanced 	<p>Properties Menu Section ^[737]</p> <p>Type-Specific Menu Section ^[737]</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> • Style • Appearance • UML Help - Displays the Enterprise Architect Help topic for this connector type • Delete - delete the connector with this option 	Advanced Menu Section ^[738] Style Menu Section ^[739] Appearance Menu Section ^[740]
Connector Role Context Menu	<p>For connectors with Roles, right-clicking a connector within up to 60 pixels of an end point displays a role-specific context menu.</p> <p>The Role context menu has three additional menu options:</p> <ul style="list-style-type: none"> • A Source/Target Role... menu option that opens the connector specification dialog with the respective role page selected • A Multiplicity submenu that enables you to set the multiplicity for the role • A Link to Element Feature menu option that displays a dialog through which you can attach the end of the connector to a specific attribute or operation 	Source Role ^[758] Connect To Element Feature ^[742]

Notes:

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.1.1 Properties Menu Section

The *Properties* section of the connector context menu contains the following options:

Menu Option	Action	Shortcut	See also
<Connector type> Properties	Open the Properties window for the selected connector		Properties Window ^[756]
Advanced	Display the Advanced menu		Advanced Menu ^[738]
Attach Note or Constraint	Attach a note or constraint to the connector		Attach a Note or Constraint ^[743]

Notes:

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.1.2 Type-Specific Menu Section

The *Type-Specific* section of the connector context menu is specific to the object, and only appears for a few different connectors. Some examples are shown below:

Connector	Menu Option	Action
Transition	Locate in State Table	Switch a State Machine diagram to a State Table, with the

Connector	Menu Option	Action
		selected Transition cell highlighted on the table
Transition	Message	Set the value of the Message
Transition	Find Triggers Associated	Identify hidden triggers, and locate a trigger in the Project Browser; if there: <ul style="list-style-type: none"> • Is one trigger, it is immediately highlighted in the Project Browser • Are two or more triggers the Element Usage dialog displays, listing the triggers; double-click on the required trigger to highlight it in the Project Browser
Association	Find Association Class	Locate and highlight the Association Class element in the Project Browser
Aggregation	Set Aggregation to Composite	Change the Aggregation to composite
Aggregation	Set Aggregation to Shared	Set the Aggregation to shared; this appears after Set Aggregation to Composite has been selected
InformationFlow	Find Information Item	(Listed only if there are Information Items on the connector) Locate the Information Item in the Project Browser; if there: <ul style="list-style-type: none"> • Is one item, it is immediately highlighted in the Project Browser • Are two or more items the Element Usage dialog displays, listing the items; double-click on the required item to highlight it in the Project Browser

Notes:

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.1.3 Advanced Menu Section

The *Advanced* section of the connector context menu contains the following options:

Menu Option	Action	Shortcut	See also
Set Source and Target	Change the source and/or target of the connector		Change the source and/or target ^[745]
Change Type	Change the connector type		Change the connector type ^[744]
Reverse Direction	Reverse the direction of the connector; for example, if the connector is an arrow, the arrowhead swaps to the other end		
Specialize Associations	Specify how the properties of this Association specialize the properties of other Associations		
Information Flows Realized	Realize any information items conveyed on an Information Flow connector between these		Information Item Realized ^[985]

Menu Option	Action	Shortcut	See also
	same two elements		Information Item Conveyed ^[984] Information Flow ^[983]
Dependency Stereotypes	Select a stereotype for the Dependency (or Trace, Role Binding, Occurrence or Represents connector)		

Notes:

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.1.4 Style Menu Section

The *Style* section of the connector context menu provides the following options:

Action	Usage	Shortcut	See also
Line Style	Set the connector line style - options are: <ul style="list-style-type: none"> Direct Auto Routing Custom Bezier Tree (Horizontal, Vertical, Lateral Horizontal or Lateral Vertical) Orthogonal (with Square or Rounded corners) 		Set the connector line style ^[745]
Pin End(s)	Pin the connector start and/or end to the current position on the target element; a sub-menu displays to offer the options of pinning the start point only, the end point only, or both Once one or both ends are pinned, a fourth option is available to unpin both ends		
Bend Line at Cursor	Insert an anchor point on the line at the point of the cursor so you can change the shape of the line	Ctrl + Q	Insert an anchor point ^[745]
Suppress Line Segment	Hide a segment of a connector so that you can view a part of the diagram that it crosses To reverse the change, right-click on the connector and select the Show All Line Segments context menu option		
Straighten Line at Cursor	Remove an anchor point on the line at the point of the cursor (This is the exact opposite of Bend Line at Cursor , and (Ctrl+Q) toggles the connector point between the options)	Ctrl + Q	Connector styles ^[745]

Notes:

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included
- If two connectors cross, the later connector has a 'hoop' at the intersection, indicating that the connectors are crossing

5.7.1.5 Appearance Menu Section

The *Appearance* section of the connector context menu provides the following options:

Menu Option	Action	Shortcut	See also
Appearance	Set the line color and line thickness of the connector		
Visibility	Set connector visibility; see the table below for sub-menu options		
Tidy Line Angles	Tidy the line angles of a custom connector		Tidy the line angles [745]

Visibility Sub-Menu:

Menu Option	Action	Shortcut	See also
Hide Connector	Hide the connector To show the connector again, follow the steps in the <i>Hide/Show Connectors</i> topic		Hide/Show Connectors [751]
Hide Connector in Other Diagrams	Hide or show the connector in other diagrams		Hide/Show Connectors [751]
Hide All Labels	Hide or show all labels attached to the connector		
Set Label Visibility	Hide or show labels attached to the connector, individually		Hide or show labels [752]

Notes:

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

5.7.2 Connector Tasks

This topic details some of the tasks associated with managing model connectors, such as:

- [Connect Elements](#) [741]
- [Connect to an Element Feature](#) [742]
- [Change Connector Styles](#) [745]
- [Arrange Connectors](#) [744]
- [Change Connector Type](#) [744]
- [Create Connector in Project Browser](#) [746]
- [Reverse Connector](#) [753]
- [Delete Connectors](#) [750]
- [Hide/Show Connectors](#) [751]
- [Hide/Show Labels](#) [752]

- [Create Generalization Set](#)^[750]
- [Change the Source or Target Element](#)^[745]
- [Set Relation Visibility](#)^[749]
- [Add a Note to a Connector](#)^[743]
- [Use Tree Style Hierarchy](#)^[755]
- [Create Connector in Project Browser](#)^[748]
- [Show Uses Arrow Head](#)^[754]
- [Set Association Specializations](#)^[753]
- [Change Sequence Message Scope](#)^[754]

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, if security is enabled, you must have **Update Element** permission to update or delete a connector

Learn More:

- [List of Available Permissions](#)^[205]

5.7.2.1 Connect Elements

Topics:

Topic	Detail	See also
Connect Elements on a Diagram	<p>The fastest and simplest ways to create connectors are using the Quick Linker and using the Toolbox</p> <p>To create another connector of the same type as the last connector you used, click on the appropriate source element and press (F3)</p>	<p>Create Connectors In Place Using the Quick Linker^[624]</p> <p>Create Connectors Using the Toolbox^[546]</p> <p>Create a Group of Elements Using UML Patterns^[1027]</p> <p>Create Domain Specific Connectors From UML Profiles^[1026]</p>
Selecting Connectors	<p>To select a connector, simply click on it; drag handles display, indicating that the connector is selected</p> <p>This gives the connector focus for keyboard commands such as (Delete), and displays connector properties in docked windows such as the Tagged Values window</p> <p>If there is more than one connector on a diagram, you can cycle through them using the arrow keys</p>	
Drag Connectors	<p>You can drag a connector to position it; click on the connector and drag it to where it is to appear - note that there are some limitations on how far or to where you can drag a connector</p> <p>You can also reattach the end of a connector to a different source or target element</p>	<p>Change the Source or Target Element^[745]</p>
Connector Properties and Commands	<p>You can double-click on a connector to change properties, or right-click to display the context menu containing commands to change connector type and direction</p>	<p>Connector Properties^[756]</p>

Topic	Detail	See also
	You can also highlight the connectors on a specific element; select the element and press (L) - all the connectors issuing from or terminating at that element are highlighted	Change Connector Type ^[744] Reverse Connector ^[753]
Create Connectors Without a Diagram	Sometimes it is useful to create relationships between elements without a diagrammatic representation; you can do this using the Project Browser and the Relationship Matrix	Add Connectors With the Project Browser ^[748] Add Connectors With the Relationship Matrix ^[500]

Notes:

- You can reposition a connector by selecting and dragging the connectors as required
- If a connector has source and target roles, you can attach either end of the connector to a specific attribute or operation in the source or target element

Learn More:

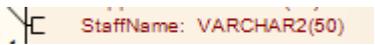
- [Connect to Element Feature](#)^[742]

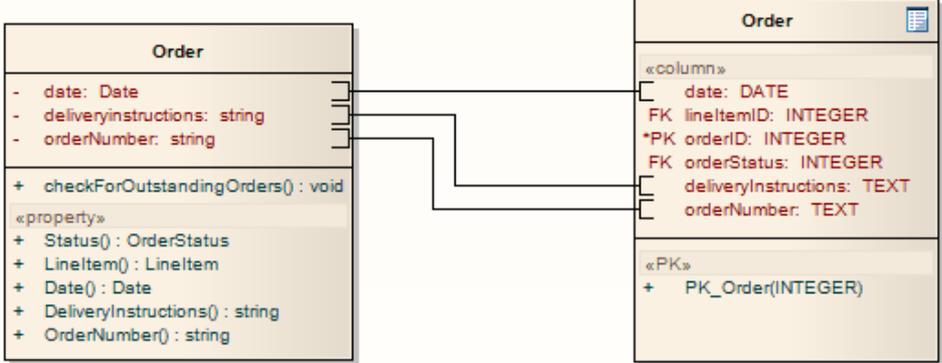
5.7.2.2 Connect to Element Feature

If a connector has source and target roles, you can connect either end of the connector to a specific operation or attribute in the source or target element. This is entirely a visual aid, to indicate which features are significant in the relationship. In code generation or transformation, the link is interpreted as a normal source-element to target-element relationship.

How to:

To connect the end of the connector to a specific operation or attribute, follow the steps below:

Step	Action	See also
1	Right-click on the end of the connector joined to the element containing the required feature	
2	Select the Link to Element Feature context menu option The Link to Element Feature dialog displays	
3	In the Feature Type field, click on the drop-down arrow and select the required feature type - Attribute or Operation The attributes or operations from the element are listed in the Feature field	
4	Click on the required attribute or operation, and click on the OK button The end of the connector changes to a bracket next to the selected feature  You might create a number of feature-to-feature relationships between two elements (such as a Class and a Table that represents the Class data) to produce a diagram similar to the following:	

Step	Action	See also
	 <p>You can change the feature to which the connector is attached by following the above procedure and selecting the new feature</p> <p>You can break the link to the selected feature in the following ways:</p> <ul style="list-style-type: none"> • Follow the above procedure, setting the Feature Type to None • Delete the attribute or operation from the element • Change the connector type to a type that does not have source and target roles • Change the connector to a different source or target element that does not contain the feature <p>Note that reversing the direction of the connector does not break the connector's attachment to the feature</p>	

5.7.2.3 Add a Note to a Connector

This topic describes how you can connect notes and constraints to graphical relationships.

Notes enable you to provide explanations and further detail for one or more connectors on a diagram, with a visible Note element.

How To:

To add a note or constraint to a connector, follow the steps below:

Step	Action	See also
1	Right-click on one of the connectors to attach a note to The context menu displays	
2	Select the Attach Note or Constraint menu option The Link Relations dialog displays	
3	Select the appropriate radio button in the Link Type panel	
4	Check all the connectors that participate in the set	
5	Click on the OK button to complete the note or constraint creation	
6	Use the normal Note dialog to enter the appropriate text for the note or constraint	

Notes:

- The constraint note is drawn slightly differently to a regular note, and has { and } automatically added to visually indicate the constraint form
- As an alternative to the procedure above, you can drag the Quick Linker arrow on an existing Note or Constraint, and drag it to the required connector; a Notelink is automatically created to attach the Note or Constraint to the connector

Learn More:

- [Quick Linker](#)^[622]

5.7.2.4 Arrange Connectors

Connectors between two elements can be moved around the element borders to create a good layout. There is a limit to how much a connector can be moved around, but generally it is very easy to find an acceptable layout. For the best layouts, use the *custom* line style; this enables you to add as many line points and bends as you require to create a clean and readable diagram.

How to:

To move and arrange connectors on a diagram, follow the steps below:

Step	Action	See also
1	Click once on the connector to select it	
2	Holding the mouse button down, move the connector in the required direction	
3	To refine the movement, click and hold very near to one end of the connector; this enables a slightly different movement range	
4	To further refine the movement and range, select either a routed, direct or custom line style; each behaves slightly differently	Connector Styles ^[745]

5.7.2.5 Change Connector Type**How To:**

To change a connector type, follow the steps below:

Step	Action	See also
1	In the Diagram View, right-click on the connector to change The context menu displays	
2	Select the Connection Detail Change Type menu option	
3	In the Connector Type field, click on the drop-down arrow and select the required connector type	
4	Click on the OK button to apply changes	

5.7.2.6 Change the Source or Target Element

After you have created a connector between two elements, you might later want to change either the source or target. Instead of deleting and re-creating the connector, Enterprise Architect enables you to change the source or target. There are two ways of doing this: using the Set Source and Target dialog or using the mouse.

How to:

To change the source or target element of a connector using the Set Source and Target dialog, follow the steps below:

Step	Action	See also
1	Right-click on the connector to open the context menu	
2	Select the Advanced Set Source and Target menu option The Set Source and Target dialog displays	
3	Click on the drop-down arrows on the From Element and To Element fields, and select the source and target elements	
4	Click on the OK button to apply changes	

To change the source or target element of a connector using the mouse, follow the steps below:

Step	Action	See also
1	Click on the connector and position the cursor over the 'handle' at one end	
2	When the cursor changes, click the mouse and drag the handle to the new element	

Notes:

- The connector does not actually move until you release the mouse button over the new source or target element; however:
 - A dotted line shows where the connector would be during the move
 - The solid outline of the nearest element or extension changes to a hatched outline as you move the cursor onto it; this helps you identify where the connector should connect to, if there are many closely-arranged elements, Parts, Ports and other extensions

5.7.2.7 Connector Styles

Connectors come in seven different routing styles. Additionally, if one connector crosses another, the second one 'hoops' over the first.

Reference:

Style	Description	See also
Direct	A straight line from element A to element B You can move the line (back and forward, up and down) to a	

Style	Description	See also
	limited degree	
Auto Routing	A vertical and horizontal route from A to B with 90-degree bends You can move the line to improve the route, but the location and number of bends are not configurable	
Bezier	A smooth curved line from A to B Bezier style is directly available for Data Flow diagram connectors, Mind Mapping connectors, State Flows, State Transitions, Object Flows, and Control Flows <ul style="list-style-type: none"> You can convert other types of relationship to Bezier style by assigning the Tagged Value <code>_Bezier</code>, with an integer value other than 0; however, some relationship types (such as Aggregate) do not accommodate this style very well This Tagged Value over-rides the value of the Style field in the connector Properties dialog 	Connector Properties 756
Custom Line	The most flexible option; you can add one or more line points and bend and push the line into virtually any shape, using the Toggle Line Point at Cursor option	
Tree Style - Vertical Tree Style - Horizontal	A line from element A to B with two right-angle bends, and the end points fixed to selected locations on the elements (Vertical or Horizontal) <ul style="list-style-type: none"> You can convert relationships to Tree Style by assigning the Tagged Value <code>_TreeStyle</code>, with a value of H (Horizontal) or V (Vertical) This Tagged Value over-rides the value of the Style field in the connector Properties dialog 	
Lateral - Vertical Lateral - Horizontal	A line from element A to B with a single right-angle bend, and the end points fixed to selected locations on the elements (Vertical or Horizontal) <ul style="list-style-type: none"> You can convert relationships to Lateral style by assigning the Tagged Value <code>_TreeStyle</code>, with a value of LH (lateral-horizontal) or LV (lateral vertical) This Tagged Value over-rides the value of the Style field in the connector Properties dialog 	
Orthogonal - Square Corner Orthogonal - Rounded Corner	You can add one or more line points and bend and push the line into a variety of shapes, using the Toggle Line Point at Cursor option All line segments thus created are either vertical or horizontal	

How to:

To set the connector style, follow the steps below:

Step	Action	See also
1	Right-click on the connector to change The context menu displays	
2	Select the Line Style option	

Step	Action	See also
3	From the submenu, select the required style: <ul style="list-style-type: none"> • Direct • Auto Routing • Custom • Tree • Lateral • (or Bezier, where appropriate) 	

Alternatively:

Step	Action	See also
1	Select the connector to change	
2	Press the following keys to change the style: <ul style="list-style-type: none"> • (Ctrl+Shift+D) for Direct • (Ctrl+Shift+A) for Auto Routing • (Ctrl+Shift+C) for Custom • (Ctrl+Shift+Z) for Bezier (where appropriate) 	

To bend a connector to quickly and easily route connectors in the required layout, follow the steps below:

Step	Action	See also
1	Right-click on the connector The context menu displays	
2	Set the line style to Custom Line (Ctrl+Shift+C); this enables the Bend Line at Cursor option in the context menu	
3	Click on the Bend Line at Cursor option to add a line point Right-clicking a line point displays the Straighten Line at Cursor context menu option, which you can use to remove the line point	
4	Using the mouse, drag the line point to the required position	

Alternatively:

Step	Action	See also
1	Hold down (Ctrl) or (Shift) and click on a point on the connector to create a line point (Ctrl)+click also removes a line point	
2	Using the mouse, drag the line point to the required position	

To tidy line angles (custom connector), follow the steps below:

Step	Action	See also
1	Right-click on the connector The context menu displays.	
2	Click on the Tidy Line Angles menu option; this nudges the custom line in horizontal and vertical increments, saving you the time of trying to get a good layout manually You can set the Tidy Line Angles option to operate by default; click on the Tools Options menu option to display the Options dialog, and select the Diagram Behavior page	

To suppress individual line segments, follow the steps below:

Step	Action	See also
1	Right-click on the connector The context menu displays	
2	Set the line style to Custom Line (Ctrl+Shift+C) This enables the Suppress Line Segment option in the context menu	
3	Click on the Suppress Line Segment option to suppress a line between two bend points The segment you right-clicked on is suppressed	
4	To show the segment again, right-click on the line and click on the Show All Line Segments context menu option One application for this is to represent the continuation of flow when your diagram crosses the page boundary marker in the Diagram View When you suppress the line segment that crosses the boundary, the link name (connector properties) displays at both ends of the hidden segment; when you print the diagram on multiple pages, the link name identifies the connection apparently broken by the page boundary	

5.7.2.8 Create Connector in Project Browser

You can create a connector from one element to another directly in the Project Browser.

How to:

To connect elements from the Project Browser, follow the steps below:

Step	Action	See also
1	In the Project Browser, either:	

	<ul style="list-style-type: none"> Right-click on the element to create a connector for, and select the Add Create Link context menu option, or Select the element, press (Insert) and select the Create Link context menu option <p>The Create Link dialog displays</p>	
2	In the Direction field, click on the drop-down arrow and select the direction of the new connector (Outgoing means this element is the source)	
3	In the Link Type field, click on the drop-down arrow and select the type of connector	
4	In the Choose target(s) list, click on the name of the target (If necessary, in the Select Target Type field click on the drop-down arrow and select a feature to list only elements having that feature)	
5	Click on the OK button to create the connector	

Notes:

- You can also reproduce an existing connector between two elements when you paste those elements from the Project Browser into a diagram as instances; an option enables you to copy just the elements, or the relationship as well

Learn More:

- [Pasting From the Tree](#)^[576]

5.7.2.9 Relationship Visibility

You can change the visibility of individual connectors or relationships, diagram by diagram.

Access: **Diagram | Advanced | Visible Relations (Ctrl+Shift+I)**

How To:

To set relationship visibility, follow the steps below:

Step	Action	See also
1	Open the diagram to change	
2	Select the Visible Relations menu option The Set Visible Relations dialog displays	
3	Select the checkbox against each list item to show, and clear the checkbox against each item to hide If you want to display the information in a more readable layout, you can resize the dialog	
4	Click on the OK button to apply the changes	

5.7.2.10 Delete Connectors

How to:

To delete a connector, follow the steps below:

Step	Action	See also
1	Right-click on the connector The context menu displays	
2	Select the Delete Connector menu option The Remove Connector dialog displays; this dialog provides the options to hide the connector so that it remains functional, or remove the connector completely	
3	Click on the appropriate radio button and click on the OK button If you select the Hide option, it has the same effect as hiding the connector on the Links tab of the source element Properties dialog, or using the Visibility Hide Connector context menu option It also hides the connector on the Relationships window	Hiding the Connector [75] ↑ Relationships [504]

Notes:

- The dialog does not display if:
 - You have previously selected the **Don't ask again** checkbox or
 - On the Links page of the Options dialog (**Tools | Options | Links**) the **Prompt on connector deletes** checkbox is not selected
- Selecting the **Don't ask again** checkbox also deselects the **Prompt on connector deletes** checkbox
- Selecting the **Prompt on connector deletes** checkbox restores the dialog if you have used the **Don't ask again** checkbox
- If you hide the dialog, the **Delete Connector** context menu option defaults to the setting you last used on the dialog; make sure that you have selected the right option to default to

5.7.2.11 Generalization Sets

A Generalization Set enables you to specify the relationship of a group of Generalizations. Each Generalization is a binary relationship that relates a specific Classifier to a more general Classifier (e.g. from a class to its superclasses). Each Generalization Set defines a particular set of Generalization relationships that describe the way in which a general Classifier (or superclass) can be divided using specific subtypes.

Access: **connector context menu | Advanced | Generalization Set | New**

How to:

To create a generalization set, follow the steps below

Step	Action	See also
1	Select the New menu option	

	The Generalization Set dialog displays	
2	In the Name field, type the name of the Generalization set; for example, Gender	
3	In the Power Type field, either: <ul style="list-style-type: none"> Type a new power type, or Click on the drop-down arrow or browser button (...) and select an existing one from the Select <Item> dialog 	Select <Item> Dialog ^[690]
4	Check the IsMember column for the child subtypes that are part of this Generalization set	

5.7.2.12 Hide/Show Connectors

Connectors/relations that appear in multiple diagrams can be selectively shown or hidden. This makes it easier to read diagrams where elements might have many connectors, but not all are relevant in the context of the current diagram.

How to:

To hide or show a connector in the current diagram, follow the steps below:

Step	Action	See also
1	Double-click on the required diagram element in the Diagram view The element Properties dialog displays	
2	Select the Links tab This lists the connectors linked to the element, whether or not they are hidden on the diagram	
3	Right-click on the connector to hide or show The context menu displays	
4	Select the Show Relation option to show the hidden connector on the diagram, or the Hide Relation option to hide the visible connector Alternatively, hide a connector by right-clicking on it on the diagram and selecting the Visibility Hide Connector context menu option; however, you must use the Links tab of the element Properties dialog to show the relationship again	

To hide or show a connector in other diagrams, follow the steps below:

Step	Action	See also
1	Right-click on the connector in the diagram The context menu displays	
2	Select the Visibility Hide Connector in Other Diagrams menu option The Set Connector Visibility dialog displays	

Step	Action	See also
3	<p>If the two connected elements have been included in other diagrams, these diagrams are listed here</p> <p>In the list, all diagrams for which the checkbox is selected show the connector; deselect the checkbox for any diagrams in which to hide the connector</p> <p>If you want to display the information in a more readable layout, you can resize the dialog</p> <p>To hide the connector in all of the diagrams listed, click on the Suppress All button</p>	
4	Click on the OK button to save the changes	

Notes:

- Certain elements, such as Requirements, do not have a Links tab in the Properties dialog

In these cases, open the Relationships window (**View | Relationships**) for the element, right-click on the relationship in the list and select the context menu option to hide or show that relationship in the diagram

Be aware that, in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions with security on, locks on the diagram and elements can make the required option unavailable

Learn More:

- [Relationships](#) ^[504]

5.7.2.13 Hide/Show Labels

How to:

To hide or display one or more labels on a connector, follow the steps below:

Step	Action	See also
1	<p>Right-click on the connector</p> <p>The context menu displays</p>	
2	<p>Select the Visibility Set Label Visibility menu option</p> <p>The Label Visibility dialog displays; if you have several long labels, you can resize this dialog for greater clarity</p>	
3	Select the checkbox against each label to display, and clear the checkbox against each label to hide	
4	Click on the OK button	

5.7.2.14 Connector In-place Editing Options

You can edit many of the Enterprise Architect connector labels directly on the diagram. Each label can be bound to a single connector field.

Topic	Detail	See also
Putting a label in Edit Mode	<p>To put a label into Edit mode, either:</p> <ul style="list-style-type: none"> • Select the Edit Label option from the context menu, or • Select a label and press (F2) <p>To save the current text to the field, either press (Return) or deactivate the Edit window</p> <p>To cancel edit mode without saving any changes, press (Esc).</p>	

5.7.2.15 Reverse Connector

You can reverse the direction of a connector without having to delete and re-create it. This is helpful if your design changes or you add the connector wrongly to begin with.

How To:

Step	Action	See also
1	Right-click on the incorrect connector	
2	Select the Connection Detail Reverse Direction context menu option	

5.7.2.16 Set Association Specializations

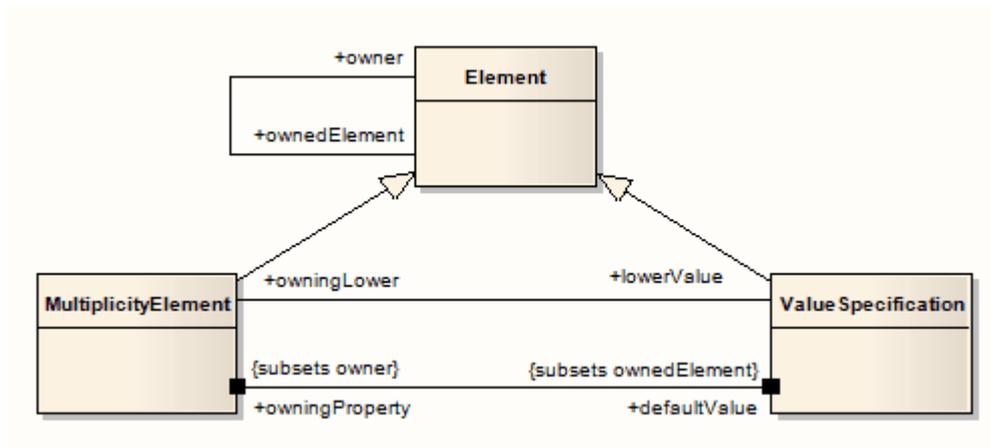
UML enables specialization of properties defined by Associations. Enterprise Architect enables this through the **Specialize Associations** option in the advanced section of the context menu for an Association.

A dialog displays, showing all Associations between the two Classes connected by the current Association and their parents.

The left two columns define the source role of the current Association, while the right two define the target role. With this you are able to select the relationships of each end of the properties listed. When a relationship is set then this is drawn at the corresponding end of the connector on any diagram it appears on.

The dialog displays when you select the **Advanced | Specialize Associations** context menu option on the lowest Association connector in the following diagram.

Example:



5.7.2.17 Change Sequence Message Scope

A message in a Sequence diagram represents a dynamic interaction from one element to another. Sometimes when you are designing your model you might have to change either the start or end point of a message as the responsibilities of elements change during design. For this reason, Enterprise Architect enables you to change the message scope by setting a new start or end element.

How To:

Step	Action	See also
1	Select the message in the Sequence diagram	
2	Right-click on the message to open the context menu	
3	Select Advanced Set Source and Target	
4	In the pop up dialog, in the From Element and To Element fields, click on the drop-down arrows and select the required elements	
5	Click on the OK button to save changes The message is re-routed to meet your changed requirements	

5.7.2.18 Show Uses Arrow Head

By default the *Use* connector in Use Cases has no arrow head.

How to:

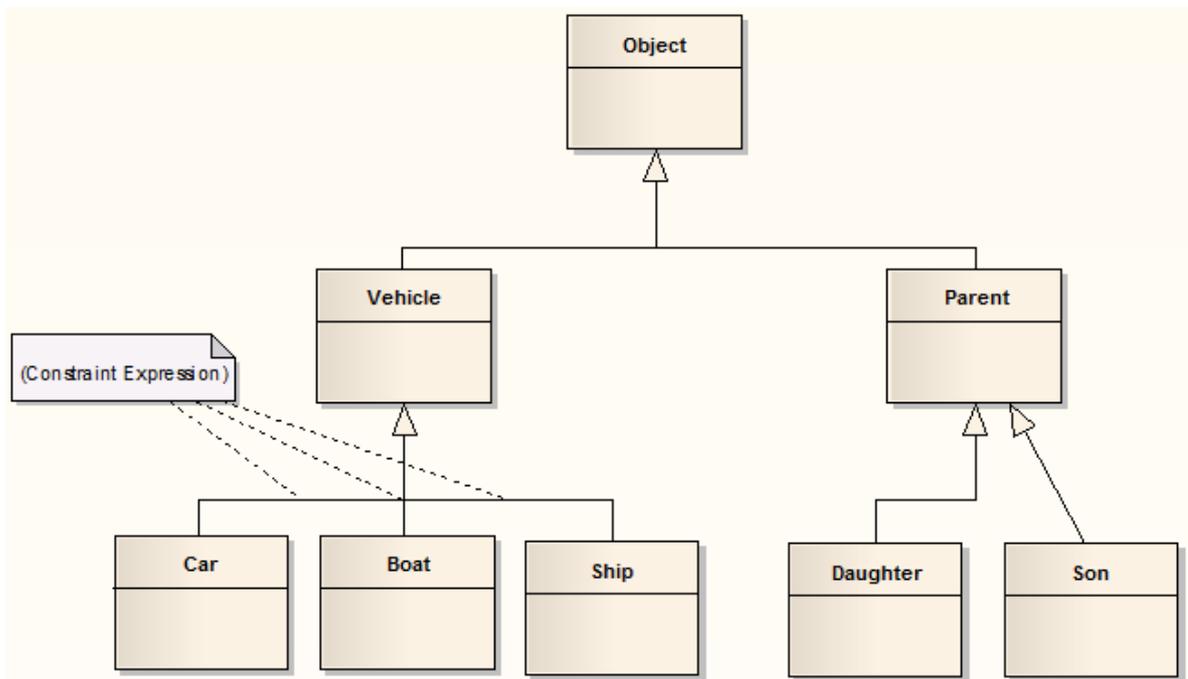
To generate arrow heads on Use connectors, follow the steps below:

Step	Action	See also
1	Select the Tools Options Links menu option The Links page of the Options dialog displays	
2	In the General panel, select the Show Uses arrowheads checkbox	

Step	Action	See also
3	Click on the Close button When you save the Use Case diagram, the Use connectors change to display arrowheads	

5.7.2.19 Tree Style Hierarchy

In Enterprise Architect you can create a tree style inheritance diagram using a special form of the *Generalization* connector, as shown below.



How To:

To create a tree style connector, follow the steps below

Step	Action	See also
1	Create a normal Generalization between two elements	
2	Right-click on the connector to open the context menu	
3	Select the Line Style Tree Style - Vertical or the Line Style Tree Style - Horizontal menu option	
4	Enterprise Architect automatically makes the Generalization layout conform to a specific shape By adding more Generalization connectors, and checking their Tree Style options, you can achieve the appearance of the diagram above You can slide the root and child Classes left and right to achieve the required result; Enterprise Architect maintains the conformity of the branch connectors	

To set this style of connector as default, follow the steps below:

Step	Action	See also
1	Select the Tools Options Links menu option The Links page of the Options dialog displays.	
2	Select the Generalization link style Default = Tree checkbox to make this branching style the default style for inheritance connectors	

Notes:

- In the example diagram, the *Son ->Parent* connector has not yet been put in **Tree Style - Vertical** style
- This style of diagram provides a clearer layout for inheritance hierarchies and is easy to work with

5.7.3 Connector Properties

To access the connector Properties dialog, double-click on a connector in a diagram. You can change several characteristics of connectors from this dialog. Many of these characteristics generate text labels on or around the connector. You can change these labels using the **Label** context menu.

The connector Properties dialog has the following pages:

- General (see below)
- Constraints
- Binding
- Source Role
- Target Role
- Tagged Values
- Advanced properties.

The General page enables you to configure the name of the connector, the direction, the line style, the stereotype (optional) and a comment.

Reference:

Field	Usage	See also
Source	Indicates in the name of the source element for the connector	
Target	Indicates in the name of the target element for the connector	
Name	(Optional) Indicates a name for the connector; if entered, the name displays on the diagram	
Alias	(Optional) Indicates an alternative name or alias for the connector	
Direction	Select the appropriate direction details: from source to destination, destination to source, or bi-directional Some connectors have arrow heads that depend on this setting; some connectors are logically dependent on this (such as Inheritance)	
Style	Select the appropriate connection style; choose from: <ul style="list-style-type: none"> • Direct 	

Field	Usage	See also
	<ul style="list-style-type: none"> • Auto-Routing • Bezier • Custom • Tree (Vertical, Horizontal, Lateral Vertical or Lateral Horizontal), or • Orthogonal (Square Corners or Rounded Corners) 	
Stereotype	<p>(Optional) Indicate the name of a stereotype for the connector, or click on the drop-down arrow and select one</p> <p>Alternatively, click on the (...) button and select the stereotype from the Stereotype Selector dialog</p> <p>If entered, the stereotype is displayed on the diagram and overrides the connector type in the RTF documentation</p>	Stereotype Selector ^[1017]
Virtual Inheritance	Indicates that inheritance is virtual; available only for <i>Generalization</i> connectors	
Scope	Select the appropriate value for the scope (used for inheritance); available only for <i>Generalization</i> connectors where the child Class is C++	
Notes	<p>(Optional) Specifies any notes on the connector; the notes are displayed in documentation, if required</p> <p>As for the Notes window, you can format the text using the Notes toolbar at the top of the field</p>	Notes Window ^[769] Notes Toolbar ^[770]

Notes:

- Generalize, Realize, Associate and Template Binding connectors also have a Binding page, which enables you to define binding expressions and parameter substitutions for the connector

Learn More:

- [Message Scope](#)^[754]
- [Constraints](#)^[757]
- [Binding](#)^[758]
- [Source Role](#)^[758]
- [Target Role](#)^[760]
- [Connector Advanced](#)^[761]
- [Tagged Values](#)^[760]
- [Template Binding](#)^[1010]
- [Label](#)^[597]

5.7.3.1 Connector Constraints

A UML connector can also have associated constraints placed on it. Constraints tell us something about the rules and conditions under which a relation operates. For example, it might be a pre-condition that a customer is of a certain type before an Association connector to an Account is allowed.

How to:

To set constraints on a connector, follow the steps below:

Step	Action	See also
1	Double-click on a connector to open the Connector Properties dialog	
2	Select the Constraints tab	
3	Fill in details of the constraint(s) that apply and click on the Save button	

Reference:

Field	Usage	See also
Constraint	The name of the constraint	
Type	Specifies the type of constraint (such as pre-condition)	
Notes	Indicates notes on the connector	
Defined Constraints	Displays the list of constraints for this connector	

Notes:

- Constraints about an Association (connector) can be added to further refine the model; constraints detail the business and operational rules for the model

5.7.3.2 Binding

Generalize, Realize, Associate and Template Binding connectors have a Binding tab on their Properties dialog. This enables you to define binding expressions and parameter substitutions for the connector.

Learn More:

- [Template Binding](#) ^[1010]
- [Parameter Substitution](#) ^[1011]
- [Parameterized Classes \(Templates\)](#) ^[943]
- [Class Details](#) ^[663]

5.7.3.3 Source Role

A connector can have certain properties assigned to one end, and be associated with the particular role that element plays in the relationship. You can enter details about this role to further develop your model.

This description refers to the role of the *Source* element in a relationship, but applies equally to the role of the *Target* element.

How to:

To set the source role details, follow the steps below:

Step	Action	See also
1	Double-click on a connector to open the Connector Properties dialog	
2	Select the Source Role tab	

Step	Action	See also
3	Enter the required details and click on the OK button	

Reference:

Field	Usage	See also
<Type> Role	Enter the name of the role to be played	
Alias	Enter an alias for the role, if required	
Role Notes	Enter notes about the role	
Derived	Indicate that the role value or values can be computed from other information	
Owned	Indicate that the role is owned by the opposite Class rather than the Association Selecting this checkbox adds a 'dot' to the appropriate end of the connector ()	
Derived Union	Indicate that the role is derived from the properties that subset it	
Multiplicity	Specify the role multiplicity (you can define the values of this field on the Cardinality tab of the UML Types dialog) This is the range of instances of the role that can be active in the relationship - for example, <i>one</i> employee can be assigned to tasks; for the target role you define the range of instances (such as tasks) the employee could be assigned to The values have the following formats: <ul style="list-style-type: none"> • *, or 0..* - zero, one or many instances • 0..n - zero or up to n instances, but no more than n • n - exactly n instances • n..* - n, or more than n instances You can also define source and target element multiplicity in the element Attribute properties	Cardinality [776] Attribute properties [698]
Ordered	Indicate that the role is a list and the list is ordered	
Allow Duplicates	Indicate that the role can contain duplicate elements (relevant only if multiplicity is > 1) This field maps to the UML property <i>isUnique</i> (selecting the checkbox maps to the <i>isUnique</i> value of <i>FALSE</i>)	
Containment	Indicate the nature of the containment at the Destination (reference, value...)	
Access	Indicate the access level for the role	
Aggregation	Indicate the type of aggregation that this role uses	
Target Scope	Indicate the level at which this role applies (instance or classifier)	
Navigability	Indicate whether or not this role is navigable (non-navigable ends are shown depending on diagram properties)	

Field	Usage	See also
Changeable	Indicate whether this role is subject to change	
Constraint(s)	Indicate a constraint on the role	
Qualifier(s)	Indicate any qualifiers or restrictions on the role; separate multiple qualifiers with a semi-colon Alternatively, click on the (...) button at the end of the field, and define a new qualifier on the Qualifiers dialog (qualifiers typed into the Qualifier (s) field are also automatically added to this dialog)	Qualifiers Dialog ^[972]
Stereotype	(Optional) Indicate the name of a stereotype that applies to this end of the Association, or click on the (...) button at the end of the field and select a stereotype from the Stereotype Selector dialog	Stereotype Selector ^[1017]
Member Type	Indicate a role type that can be used when generating collection Classes for multiplicity > 1	

Notes:

- Source role details are displayed at the start end of a connector; if you have drawn the connector the wrong way, you can always use the **Reverse Direction** menu option from the connector context menu

5.7.3.4 Target Role

A connector can have certain properties assigned to one end, and be associated with the particular role that element can play in the relationship.

You can enter details about this role to further develop your model.

How to:

To set the destination role details, follow the steps below:

Step	Action	See also
1	Double-click on a connector to open the Connector Properties dialog	
2	Select the Target Role tab	
3	The details and appearance of this tab are identical to the Source Role tab	Source Role ^[758]

Notes:

- Destination role details are displayed at the terminating end of a connector on the diagram

5.7.3.5 Connector Tagged Values

The Tagged Values tab of the connector Properties dialog simply provides the Tagged Values window within the frame of the Properties dialog.

You can define Tagged Values for the connector and, on Association and Aggregation connector types, you can set additional Tagged Values for the source and/or target role.

How to:

To set Tagged Values for the connector, follow the steps below:

Step	Action	See also
1	On the Properties dialog for the connector, click on the Tagged Values tab	
2	Select the connector type, Connector Source or Connector Target as required	
3	Either click on the New Tags button or press (Ctrl+N) The Tagged Value dialog displays	
4	In the Tag field type the tag name and value, or click on the drop-down arrow and select a predefined Tagged Value type	
5	Click on the OK button to save the changes	

Learn More:

- [Tagged Values](#) 

5.7.3.6 Connector Advanced

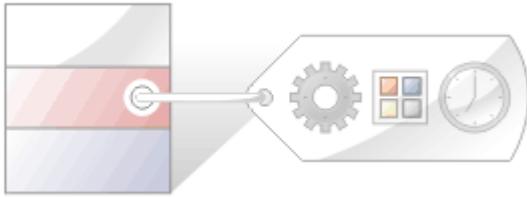
The Advanced page displays the advanced properties of the connector (where they exist) and enables you to reset the values of these properties.

How to:

To set a value for an advanced (or custom) property, follow the steps below:

Step	Action	See also
1	Click on the data field to the right of the property name	
2	Depending on the property, either: <ul style="list-style-type: none"> • Type the value in free text • Click on the drop-down arrow and select the value from the list • Click on the (...) Browse button and search for the required value 	
3	Click on the OK button	

5.8 Tagged Values



Topics:

Topic	Detail	See also
What is a Tagged Value	<p>Tagged Values are a convenient way of adding additional information to an element, beyond what is directly supported by UML. The UML specification provides the Tagged Value element for just this purpose. Often Tagged Values are used during code generation or by other tools to pass information or operate on elements in particular ways. Tagged Values are the preferred method of extending the code generation capabilities of the modeling tool per element / per language.</p> <p>A Tagged Value, strictly, is the value of a property of a modeling item, the property being called a tag. For example, a Class element called Person might have a tag called Age with the Tagged Value of 42. The combination of tag and value, however, is often referred to as a Tagged Value.</p> <p>Enterprise Architect defines a Tagged Value Type, which constrains the possible values of a tag and can specify how a value is assigned to the tag. For example, the tag Age might have a Tagged Value Type of Integer, so the user simply types in a numeric value. Alternatively, the type could be Spin, with lower and upper limits of, say, 20 and 120, so the user sets a value by clicking on arrows in the field to increment or decrement the value within the limits of 20 and 120.</p> <p>To quickly add Tagged Values to one of more elements, see the Quick Add - Tagged Value to Elements topic.</p>	Tagged Value Type ^[775] Quick Add - Tagged Value to Elements ^[763]
The Tagged Values Window	<p>The Tagged Values window is used to view and modify Tagged Values for the currently selected modeling item, either in the current diagram or in the Project Browser.</p> <p>The Tagged Values window is a dockable window. You can use it to perform the following actions:</p> <ul style="list-style-type: none"> • Assign a Tagged Value to an Item • Modify Tagged Values • Assign Notes to a Tagged Value • Perform advanced tag management <p>A Technology Developer can also create new structured Tagged Values, reference data Tagged Values and custom Tagged Values from predefined Tagged Value Types.</p>	Assign a Tagged Value to an Item ^[764] Modify Tagged Values ^[766] Assign Notes to a Tagged Value ^[765] Perform Advanced Tag Management ^[767] Custom Tagged Values ^[1113] Tagged Value Types ^[1108]
Model Elements and Features	<p>The following model components can use the Tagged Values window as a convenient way to quickly view and modify Tagged</p>	

Topic	Detail	See also														
with Tagged Values	<p>Values:</p> <table border="1" data-bbox="491 360 1230 904"> <thead> <tr> <th data-bbox="491 360 740 416">Component</th> <th data-bbox="740 360 1230 416">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="491 416 740 506">Elements</td> <td data-bbox="740 416 1230 506">Elements display their own Tagged Values and inherited values.</td> </tr> <tr> <td data-bbox="491 506 740 595">Object Instances</td> <td data-bbox="740 506 1230 595">Object Instances display owned tags and those inherited from their classifier.</td> </tr> <tr> <td data-bbox="491 595 740 707">Ports and Parts</td> <td data-bbox="740 595 1230 707">Ports and parts display information similar to that of a Port/Part 'Type' instead of a classifier. Tags are inherited from parents and other structures of the Ports type.</td> </tr> <tr> <td data-bbox="491 707 740 797">Attributes</td> <td data-bbox="740 707 1230 797">Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.</td> </tr> <tr> <td data-bbox="491 797 740 842">Operations</td> <td data-bbox="740 797 1230 842">Owned properties only.</td> </tr> <tr> <td data-bbox="491 842 740 904">Connectors</td> <td data-bbox="740 842 1230 904">Owned properties only.</td> </tr> </tbody> </table> <p>When over-riding an inherited property, Enterprise Architect copies the tag from the parent down to the child element and sets the new value, leaving the original tag unchanged.</p>	Component	Description	Elements	Elements display their own Tagged Values and inherited values.	Object Instances	Object Instances display owned tags and those inherited from their classifier.	Ports and Parts	Ports and parts display information similar to that of a Port/Part 'Type' instead of a classifier. Tags are inherited from parents and other structures of the Ports type.	Attributes	Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.	Operations	Owned properties only.	Connectors	Owned properties only.	
Component	Description															
Elements	Elements display their own Tagged Values and inherited values.															
Object Instances	Object Instances display owned tags and those inherited from their classifier.															
Ports and Parts	Ports and parts display information similar to that of a Port/Part 'Type' instead of a classifier. Tags are inherited from parents and other structures of the Ports type.															
Attributes	Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.															
Operations	Owned properties only.															
Connectors	Owned properties only.															
Tagged Values Toolbar Buttons	<p>The buttons in the Tagged Values toolbar enable you to add, edit, sort, delete and arrange the Tagged Values of model features</p>  <p>From left to right, the button functions are as follows:</p> <ul style="list-style-type: none"> • The Show in compartments button displays the Tagged Values in element compartments on diagrams • The Sort and Show Alphabetically button sorts the current Tagged Values for the element alphabetically • The New Tagged Value button adds a new tag, to which you assign a value • The Edit Tagged Value Notes button enables you to create notes that explain the purpose of the Tagged Value • The Delete Tagged Value button removes the currently selected Tagged Value • The Default Tagged Value Types button enables quick access to tag definitions created in the Configuration menu • The Tagged Value Options button enables you to show or hide the fully qualified paths for the Tagged Values in the window, and to show duplicate Tagged Values • The Help button displays help relating to use of the Tagged Values window 															

5.8.1 Quick Add - Tagged Value To Elements

You can add a Tagged Value to one or more elements with a special shortcut. You can also use the Current Element toolbar. The last button is a shortcut to the **Add Tagged Value** function.

To delete this property you must open the element Properties dialog, go to the Tagged Values tab and

manually delete the item. There is currently no shortcut to delete tags from multiple elements simultaneously.

To add notes to the Tagged Value, go to the Tagged Values tab, click on the Tagged Value name, and click on the **Edit Notes** button in the tab toolbar. The Notes dialog displays.

Any Notes text you enter also displays in the Info section at the bottom of the Tagged Values window.

Access: **Add | Tagged Value**

Use to:

- Add Tagged values to elements

How To:

Step	Action	See also
1	From an element context menu (or the context menu of a multi-selection) choose the Add Tagged Value menu option (Alternatively, select one or more elements and press (Shift+Ctrl+T)) The Tagged Values dialog displays, which enables you to enter a Name and Value for the tag	
2	Click on the OK button to add your new Tagged Value to all the currently selected elements	

5.8.2 Assign a Tagged Value to an Item

You can assign Tagged Values to several model features.

Access: **View | Tagged Value**

Use to:

- Assign a Tagged Value to an item

How To:

Step	Action	See also
1	Select the model feature to associate with the defined Tagged Value	
2	Ensure that the Tagged Values window is visible (select the View Tagged Values menu option, or press (Ctrl+Shift+6)) Alternatively, open the Properties dialog for the object and select the Tagged Values tab	
3	Either click on the New Tags button  or press (Ctrl+N) The Tagged Value dialog displays	

Step	Action	See also
4	In the Tag field, type the tag name or click on the drop-down arrow and select the appropriate tag to assign to the item	
5	If appropriate, type a specific value for the tag in the Value field	
6	To confirm selection of the Tagged Value, click on the OK button	

Learn More

- [Predefined Tagged Value Types](#)^[1108]
- [Assign Notes to a Tagged Value](#)^[765]
- [Modify Tagged Values](#)^[766]
- [Model Elements and Features with Tagged Values](#)^[762]

5.8.3 Assign Notes to a Tagged Value

Once a Tagged Value has been assigned to a model feature, you can add information and notes describing the Tagged Value to the information property of the Tagged Value.

Access: **View | Tagged Values** (Ctrl+Shift+6)

Use to:

- Assign a note to a Tagged Value

How To:

Step	Action	See also
1	Select the Tagged Values menu option The Tagged Values window displays	
2	Click on the model feature for which to edit the Tagged Values Its Tagged Values display in the Tagged Values window	
3	Click on the Tagged Value to add information to	
4	Click on the Edit Tagged Value Notes button or press (Ctrl+E) The Tagged Value Note dialog displays	
5	In the Note field, type the information relating to the Tagged Value Click on the OK button The information is displayed in the lower portion of the Tagged Values window whenever the Tagged Value is selected	

Learn More:

- [Model Features](#)^[762]

5.8.4 Modify Tagged Values

Once a Tagged Value has been assigned to the model feature it is possible to edit the values from the Tagged Values window.

Use to:

- Edit Tagged Values

How To:

Step	Action	See also
1	Click on the View Tagged Values menu option, or press (Ctrl+Shift+6) The Tagged Values window displays	
2	Click on the model feature for which to edit the Tagged Values The window shows all of the tags for the selected feature, each with their current value	
3	Edit the fields as appropriate; the information entered can only reflect the value types that have been defined by the tag's Tagged Value Type There are four types of value field for a Tagged Value: <ul style="list-style-type: none"> • 'Open' fields', in which you can type any appropriate value • 'Drop-down list' fields, where you click on the drop-down arrow to select from a discrete list of possible values such as M or F, or Win, Lose or Draw • 'Spin' fields, where you click on up or down arrows in the field to increase or decrease the value within certain limits • 'Further detail' fields, where you click on an ellipsis (...) in the field to display a dialog in which you enter information (such as notes) or indicate a source of further information (such as a classifier) 	Instance Classifier 690

Notes:

- To override a Tagged Value defined in a parent element, edit the value in the from <parentname> compartment of the Tagged Values window

Once this has been done the tag is moved into the selected element's Tagged Values; this does not affect the Tagged Values defined in the parent element

5.8.5 Show Duplicate Tags

Tagged Values are by default set to hide duplicate values. This setting is used to facilitate inherited and overridden tag names.

How To:

To set the Tagged Values window to show duplicate values, follow the steps below:

Step	Action	See also
1	On the Tagged Values window toolbar, click on the Options icon	

Step	Action	See also
2	Select the Show Duplicate Tags context menu option	

Alternatively:

Step	Action	See also
1	Select the Tools Options menu option The Options dialog displays	
2	From the hierarchical tree, select the Objects item	
3	Select the Show Duplicate Tags checkbox	

Notes:

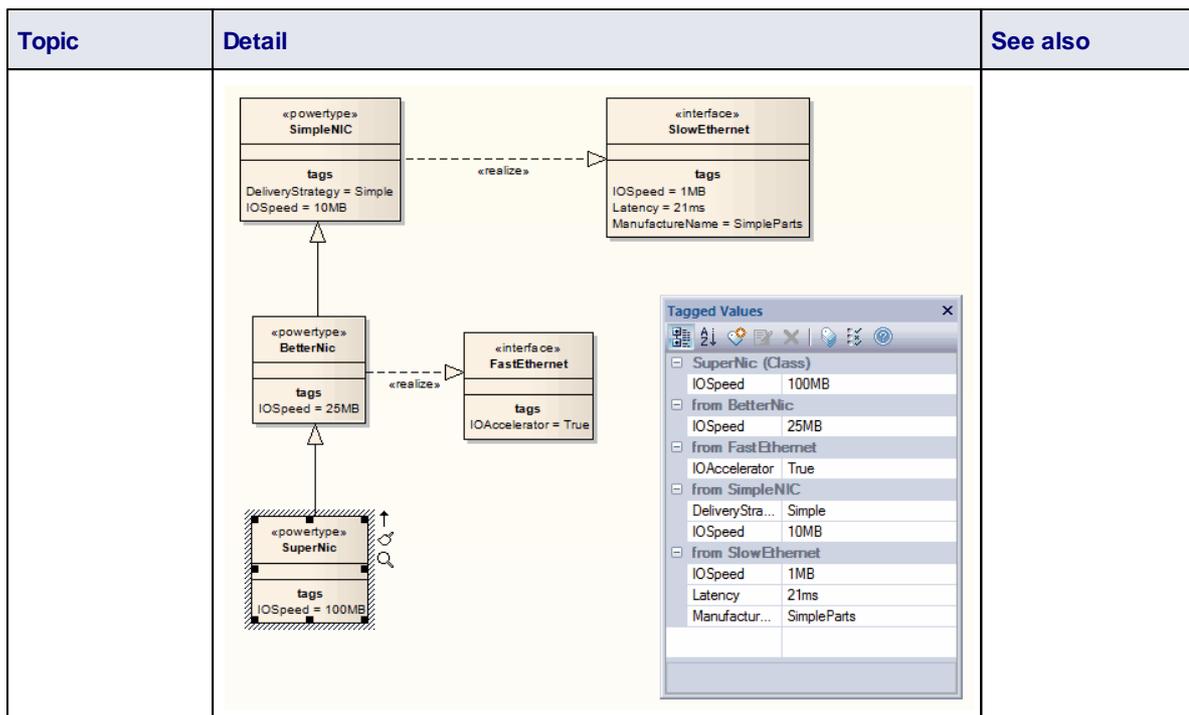
- In either procedure, to hide duplicate values again deselect the option or check box

5.8.6 Advanced Tag Management

Tagged Values can also be managed within a type hierarchy and with respect to element instances, using the Tagged Values window.

Topics:

Topic	Detail	See also
Usage	Using the Tagged Values window it is possible to: <ul style="list-style-type: none"> • View Tagged Values inherited from parent Classes or realized interfaces or applied stereotypes • Override Tagged Values derived from parents or applied stereotypes with a unique value for the current element • Delete Tagged Values from the current element (if a parent version of the Tagged Value exists, it re-appears in the list after the override is deleted). 	
Tag Hierarchy	The diagram below illustrates a complex tag hierarchy and the way Tagged Values can be either inherited or overridden in specialized Classes to create the final tagged property set for an element. Note also that a similar concept applies to instances, in which case the full tag set is created from the directly owned tags, plus all of those merged in from the classifier's type hierarchy, additional stereotypes and realized interfaces.	

**Learn More:**

- [Tagged Values](#) ⁷⁶²

5.9 Notes

Notes are the main documentation feature you use to describe an element, diagram, feature or connector; in the documentation that Enterprise Architect generates, notes feature prominently. You can record and view notes on a modeling object using the Notes window.

Access: **View | Notes (Ctrl + Shift + 1)**

Topics:

Topic	Detail	See also
<p>Usage</p>	<p>You use the Notes window to view and edit the documentation (notes) associated with elements, diagrams, attributes, operations and connectors, either from a diagram or from the Project Browser</p> <p>When you select an element, the note displayed changes to reflect the current selection; if you make changes to notes in this window, they are saved</p> <p>If you want to display the Notes information in a more readable layout, you can resize the window</p> <p>You can also format the notes text using the Notes toolbar at the top of the Notes window</p> <p>You can cut, copy, paste and delete text in the Notes window, or in any Notes or Description field that shows the Notes toolbar, using a right-click context menu; from the context menu, you can also select an option to spell-check a highlighted word</p> <p>You can also edit notes by double-clicking on an element or connector in a diagram or in the Project Browser, to open the Properties dialog; any formatting changes made in one display are reflected in the other</p> <p>On the Testing, Maintenance and Project Management windows, any descriptive, history, input or results text for a selected item is also displayed in the Notes window; you cannot edit this text in the Notes window</p>	<p>Notes Toolbar^[770]</p>
<p>Glossary Entries</p>	<p>The Notes window or field also enables you to create a Project Glossary entry from text you have highlighted in the window or field</p> <p>To create the Glossary entry, follow the steps below:</p> <ul style="list-style-type: none"> • Highlight the notes text to use as the Glossary definition (if suitable text is available), and press (Ctrl+C) to copy it • Highlight the text to use as the Glossary term, and right-click on it to display the context menu • Select the Create Glossary Definition menu option; the Glossary Detail dialog displays, with the selected term in the Term field • If you have copied some definition text, paste it into the Meaning field; otherwise type a suitable definition of the term in this field • In the Type field, select the appropriate term type • Click on the Apply button to save the new Glossary 	<p>Project Glossary^[362]</p> <p>Glossary Detail^[362]</p>

Topic	Detail	See also
	<p>definition</p> <p>The term displays in the Notes text as a roll-over hyperlink which, when you move the cursor over it, displays the Glossary definition of the term</p> <p>Having created a glossary definition anywhere else in the model, you can insert the glossary term in the text of the Notes window (or Notes panel of a dialog) as a rollover hyperlink to the definition</p>	
Inserting a Glossary term in a Notes Window	<ul style="list-style-type: none"> • In the Notes window, move the cursor to the point in the text at which to insert the glossary term • Press (Ctrl+Space); the glossary term selection list displays • Double-click on the term to insert in the Notes text; the term is inserted as a rollover hyperlink to the definition 	
Additional keyboard shortcuts	<ul style="list-style-type: none"> • Insert Date and Time (F5) • Select line of text (F8) • Spell check Notes text only (F7) • Undo changes (Ctrl+Z) • Redo changes (Ctrl+Y) or (Ctrl+Shift+Z) • Copy (Ctrl+C) • Paste (Ctrl+V) • Cut (Ctrl+X) <p>Any Note text appearing in the element Note compartments in diagrams is not formatted</p>	

5.9.1 Notes Toolbar

Although it is not an independent toolbar that you can pin to the screen top or sides, or float in your work area, the Notes toolbar appears in many places across Enterprise Architect in the **Notes**, **Description** and **History** fields.

Topics:

Topic	Detail	See also
Where the Toolbar Appears	<p>Throughout Enterprise Architect on various tabs and pages of, for example, the:</p> <ul style="list-style-type: none"> • Element and Connector Properties dialogs • Operations and Attributes Properties dialogs • Diagram Properties dialog • Hyperlink Details dialog • Testing window descriptions • Notes window • Glossary Detail dialog • Assigned Resources dialog • Project Management window • Scenarios & Requirements window 	
Options		<p>Hyperlinks [1292]</p> <p>Linked</p>

	<p>The options of this toolbar operate on selected text and any new text continuing from the formatting</p> <p>The options (with some keyboard shortcuts) are, from left to right:</p> <ul style="list-style-type: none">• Make text bold (Ctrl+B)• Make text italic (Ctrl+I)• Underline text (Ctrl+U)• Change the font color of the text• Insert list bullet points (Ctrl+.) (full stop)• Insert list numbering (Ctrl+1)• Make text superscript• Make text subscript• Insert a hyperlink - this displays the Hyperlink Details dialog, on which you specify the type of hyperlink and type in or browse for the location of the target of the hyperlink; you can use full paths or local (path substitution) paths• Create a new linked document for the element, or edit an existing one; if the element does not have a linked document, the New Linked Document dialog displays	Documents [730]
--	---	------------------------------------

Notes:

- If the toolbar is displayed but grayed out, the text field is read-only and cannot be edited; other long text fields in Enterprise Architect might not have the toolbar, in which case the Notes facility is not available for those fields
- For any Notes text that is displayed on a diagram, you must select the **Render Formatted Notes** checkbox on the Feature Visibility dialog in order to reproduce the formatting
- You can create a Project Glossary term and definition from text in any field that has the Notes toolbar

Learn More:

- [Feature Visibility dialog](#) [585]
- [Project Glossary](#) [362]
- [Notes](#) [769]

5.10 Reference Data



Reference data is used in many places to provide content for drop-down list boxes. Setting up a project often involves setting up the base set of reference types to use. Reference data options can be set up from the **Settings** menu, including:

- **People**
- **General Types**
- **Maintenance**
- **Metrics and Estimation**
- **UML**
- **Data Types**

Having set up the reference data in a project, you can also export and import it between projects.

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Manage Reference Data - Update permission to update and delete reference items.

Learn More:

- [People](#) ^[778]
- [General Types](#) ^[782]
- [Maintenance](#) ^[787]
- [Metrics and Estimation](#) ^[787]
- [UML](#) ^[772]
- [Data Types](#) ^[777]
- [Export and Import](#) ^[236]
- [Manage Reference Data - Update](#) ^[205]
- [Applying Stereotypes](#) ^[1017]
- [Custom Stereotypes](#) ^[1039]

5.10.1 UML Types

The **UML Types** dialog enables you to configure stereotypes, Tagged Value types and the cardinality list for your project.

Access: **Settings | UML Types**

Use to:

- Add, modify and delete stereotypes
- Customize stereotypes appearance
- Updating shape scripts and metafiles associated with Stereotypes
- Create/modify tagged values
- Define cardinality rules

Learn More:

- [Stereotypes](#) ^[773]
- [Tagged Value Types](#) ^[775]
- [Cardinality](#) ^[776]

5.10.1.1 Stereotype Settings

Enterprise Architect has an extensive set of Standard Element Stereotypes that you can apply to any UML construct. Using the **Stereotypes** tab of the **UML Types** dialog, a Technical Developer can also customize the stereotypes for your project by adding, modifying and deleting them. For information on customizing stereotypes, see Custom Stereotypes.

Stereotypes can be modified to make use of metafiles (image files) or customized colors, or to make use of the Enterprise Architect Shape Scripts to make new element shapes to determine the shape and dimensions of the element.

Access: [Settings](#) | [UML Types](#) > [Stereotypes](#)

Use to:

- Add, modify and delete stereotypes
- Customize stereotypes appearance
- Updating shape scripts and metafiles associated with Stereotypes

Reference:

Field	Usage	See also
Stereotype	Specify the name of the stereotype.	
Group name	Enable grouping of stereotype features by a plural name, for attributes and operations, which is shown on diagrams in the attribute and operations compartments.	
Base Class	Enable the stereotyped element to inherit the base characteristics from a pre-existing element type.	
Notes	Indicates any notes concerning the stereotype (not the elements to which the stereotype is to be applied).	
None	Switch to the default element appearance.	
Metafile	Enable an image file to be used for the appearance of the stereotype.	
Shape Script	Specify custom shapes for the stereotype using the Enterprise Architect Shape Scripting language. For more information see the Shape Scripts topic.	Shape Scripts ^[1089]
Assign	Add the associated metafile or Shape Script from the stereotyped element.	
Remove	Remove the associated metafile or Shape Script from the stereotyped element.	
Fill	Set the default background color of the element.	

Field	Usage	See also
Border	Control the border color.	
Font	Control the color of the stereotype font.	
Reset	Reset the appearance of the element to the default element appearance.	
Preview	Provides a visual representation of the currently active Stereotype.	
New	Enables a new stereotype to be created.	
Save	Saves the Stereotype details and adds the Stereotype to the Defined Stereotypes list.	
Delete	Removes a Stereotype from the Defined Stereotypes list.	

Learn More:

- [Standard Element Stereotypes](#) ^[1019]
- [Applying Stereotypes](#) ^[1017]
- [Custom Stereotypes](#) ^[1039]

5.10.1.1.1 Shape Editor

The **Shape Editor** enables a Technology Developer to specify custom shapes via a scripting language; that is, to create *Shape Scripts*. These custom shapes are drawn instead of the standard UML notation. Each script is associated with a particular Stereotype, and is drawn for every element of that stereotype.

Access: [Settings](#) | [UML Types > Stereotypes : Assign](#)

Use to:

- Create a custom shape

Reference:

Field	Usage	See also
Format	The format in which the script is written.	
Import	Opens a Windows browser dialog, allowing for a script file to be loaded.	
Export	Opens a Windows browser dialog, allowing for a script file to be saved.	
Edit window	The area in which a script can be written.	
Preview of main	A visual representation of the compiled script. Nothing displays until the Refresh button is clicked.	
Next Shape	If a composite shape is defined within the Edit Window , clicking on the Next Shape button pages through the components of the shape.	

Refresh	Parses your script and produces a visual example of the shape in the Preview of main window.	
OK	Exit from the Shape Editor; don't forget to save your scripts from the Stereotype tab.	Getting Started With Shape Scripts ^[1090]

Notes:

- Shape Scripts adopt the same color gradient settings as normal elements, as defined in the **Standard Colors** page of the **Options** dialog
- If an element's appearance is modified by a Shape Script, many of the Advanced context menu options for that element are disabled
- Once you have finishing writing your Shape Script, click on the **OK** button. To save the Shape Script you must click on the **Stereotypes** tab

Learn More:

- [Writing Scripts](#) ^[1091]
- [Advanced Settings](#) ^[649]

5.10.1.2 Tagged Value Types

Tagged Values are used in a variety of places within Enterprise Architect to specify additional information about an element or connector. The Tagged Value Types tab of the UML Types dialog enables a Technology Developer to rapidly create Tagged Values, using a range of predefined structured Tagged Values to create structured tags that adhere to a specific format. For example, for model features that use the predefined tag Boolean you can use the Tagged Values window to assign a value of *True* or *False* and no other value.

You can also add default Tagged Value names and create predefined reference data Tagged Value types and custom masked Tagged Value types.

Any Tagged Value names created display in the drop-down lists of Tagged Value names in the Tagged Value dialogs for elements, operations and attributes.

Access: **Settings | UML Types > Tagged Value Types**

Use to:

- Create new tagged values

Reference:

Field	Usage	See also
Tag Name	Type the new name of the Tagged Value.	
Description	Type a description of the Tagged Value.	
Detail	Type any additional information necessary.	
New	Click to begin creating a new Tagged Value.	
Save	Click to save the details of a Tagged Value.	
Delete	Click to delete a Tagged Value from the Defined Tag Types list.	

Field	Usage	See also
Defined Tag Types	A list of previously defined Tagged Values.	

Notes:

- You can transport these Tagged Value Type definitions between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu. Tagged Value Types are exported as *Property Types*

Learn More:

- [Predefined Structured Tagged Values](#) ^[1108]
- [Create Structured Tags](#) ^[1111]
- [Predefined Reference Data Tagged Value Types](#) ^[1113]
- [Custom Masked Tagged Value Types](#) ^[1113]
- [Tagged Values Window](#) ^[762]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

5.10.1.3 Cardinality

The **Cardinality Values** tab of the **UML Types** dialog enables you to add, modify and delete values in the default cardinality list.

The cardinality values are used to define the multiplicity of source and target elements in relationships. This is the range of instances of the role that can be active in the relationship; for example, one employee can be assigned to tasks; for the target role you define the range of instances (such as tasks) the employee could be assigned to.

The cardinality values are also used to define the multiplicity of a Classifying element; that is, the number of instances of the element that can exist. For example, the Class element *Building Walls* might have a multiplicity of 2..n, meaning that at least two walls must exist (to support the roof) but there can be many walls if the building design required it.

The values have the following formats:

- ***, or **0..*** - zero, one or many instances
- 0..n** - zero or up to n instances, but no more than n
- n** - exactly n instances
- n..*** - n, or more than n instances.

Access: **Settings | UML Types > Cardinality Values**

Use to:

- Create Cardinality values

Reference:

Field	Usage	See also
Cardinality	Type the new name of the Cardinality Value.	

Field	Usage	See also
New	Type a description of the Cardinality Value.	
Save	Saves the Cardinality value and adds it to the Cardinality List .	
Delete	Deletes a Cardinality value from the Cardinality List .	
Cardinality List	A list of already defined Cardinality values.	

Notes:

- You can transport these cardinality values between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]
- [Source Role](#) ^[758]
- [Destination Role](#) ^[760]

5.10.1.4 Data Types

Different programming languages support different inbuilt data types. The **Programming Languages Datatypes** dialog enables you to extend and manage the set of inbuilt data types associated with a language as well as create new programming languages for use within Enterprise Architect.

Access: **Settings > Code Datatypes**

Use to:

- Create new programming languages to use within Enterprise Architect
- Extend and manage inbuilt data types

Reference:

Field	Usage	See also
Product Name	Specify the name of the programming language.	
Add Product	Add a new programming language to the drop-down fields for Class elements within the Enterprise Architect model and enable the new language to be made available to the Code Template Editor once at least one datatype has been added to the language.	Code Template Editor ^[1485]
Datatype	Specify the name of the datatype; this is the language-specific name of the datatype.	
Common Type	Specify the common type, the generic name of the datatype; for example, the Java <i>boolean</i> datatype has a common datatype <i>Boolean</i> .	
New	Create a new data type.	
Save	Save the newly created datatype.	

Field	Usage	See also
Delete	Delete the selected datatype.	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Configure Datatypes permission to update and delete data types
- You can delete data types that you have defined, but you cannot delete any of the predefined data types
- You can transport these data types between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]
- [Permission List](#)^[758]

5.10.2 People

Access: **Settings | Project Types | People**

Use to:

- Maintain Authors involved within a project
- Define role types that are captured within Enterprise Architect
- Record information on project resources
- Capture client details associated with the current model

Learn More:

- [Project Authors](#)^[778]
- [Project Roles](#)^[780]
- [Project Resources](#)^[781]
- [Project Clients](#)^[782]

5.10.2.1 Project Authors

You can define the people who are working on a project, such as the authors of specific elements.

Access: **Settings | Project Types | People > Project Authors(s)**

Use to:

- Maintain Authors involved within a project

Reference:

Field	Usage	See also
Name(s)	Specifies the name of the person registered as a Project Author.	Select Users ^[779]

	<p>If you are using a Windows Active Directory, you can select names from the directory. Click on the (...) (Browse) button to display the Select Users dialog.</p> <p>You can also indicate a list of names separated by semi-colons. This enables you to define a group of people sharing a role, such as a team of Developers, Testers or Analysts. Do not leave any spaces between the names and the semicolon.</p>	
Role	<p>Specifies the role the Project Author plays in the project (such as Designer, Analyst, or Architect).</p> <p>You can specify a role name or click on the drop-down arrow and select a role defined through the Project Roles tab.</p>	Project Roles ^[780]
Notes	Specify any additional notes concerning the Project Author.	
New	Add further Authors.	
Save	Add the new Author to the Defined Authors list.	
Delete	Removes an Author from the Defined Authors list.	
Defined Authors	Review the Project Authors already defined.	

Notes:

- If you enter multiple names, Enterprise Architect adds them separately and in alphabetical order to the **Defined Authors** list. If you then click on one of these names, Enterprise Architect displays that name only in the **Name(s)** field
- If you type a role, this is not added to the roles or the **Project Roles** tab
- You can transport these author definitions between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.10.2.1.1 Select Users

If your company is using a Windows Active Directory, you can select the Project Author names from the local or corporate-wide directory.

Access: **Settings | Project Types | People > Project Authors(s) : (...)**

Use to:

- Select Project Authors from a Windows Active Directory directory

Reference:

Field	Usage	See also
Object Types...	Opens the Object Types dialog, this dialog provides a choice of object types that can be used.	

Locations...	Defines the root location from which to begin a search.	
Check Names	Matches object names listed in the Enter the object name to select section	
Enter the object names to select	Specify object names to search for.	
Advanced...	Opens the Advanced dialog to provide further search options. For more information on this dialog, please refer to Windows Help and Support	

Notes:

- Multiple Entries can be typed into the **Enter the object names to select** section, ensure that each object is separated with a semicolon (Example: Name01; Name02)

5.10.2.2 Project Roles

People associated with a project play a *role* in analysis, design or implementation, such as Application Analyst, Architect, Developer and Project Manager. Project roles define the activities that resources can undertake.

Access: [Settings](#) | [Project Types](#) | [People](#) > [Project Roles](#)

Use to:

- Define role types that are captured within Enterprise Architect

Reference:

Field	Usage	See also
Role	Indicates the name of the role.	
Description	Indicates a description of the role.	
Notes	Specifies any additional information related to the role.	
New	Add further Roles.	
Save	Add the new role to the Defined Roles list.	
Delete	Removes a role from the Defined Role list.	
Defined Roles	Review the Project Roles already defined. The Defined Roles list is available for selection for any element in the model; for example, you can select roles on the Project Authors tab of the People dialog, and the Resource Allocation tab of the Project Management window. You can also specify other roles on these dialogs, but such roles are not added to the Defined Roles list.	Project Authors ^[778] , Resource Allocation ^[349]

Notes:

- Deleting a role has no effect on any Project Author definition having this role; the deleted role becomes a simple text entry in the Project Author definition

- You can transport these role definitions between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.10.2.3 Project Resources

Resources are, for example, project authors, analysts, programmers and architects. That is, anyone who might work on the system over time, either adding to the model or programming and designing elements of the system outside Enterprise Architect.

Access: **Settings | Project Types | People > Project Resources**

Use to:

- Record information on project resources

Reference:

Field	Usage	See also
Name	Indicates the name of the person listed as a resource. The resource name is available for use in Resource Management.	Resource Management ^[349]
Organization	Indicates the name of the organization employing the resource.	
Role(s)	Indicates the role the resource plays in the project (for example, Designer, Analyst, Architect).	
Phone 1, Phone 2, Mobile, Fax	Indicates the contact telephone numbers for the resource.	
Email	Indicates the email address for the resource.	
Notes	Indicates any additional notes on the resource.	
Available Resources	Review resources that have already been defined.	
New	Adds further resources.	
Save	Adds a new resource to the Available Resource list.	
Delete	Deletes a resource from the Available Resources list.	

Notes:

- You can transport these resource definitions between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#)^[237]

- [Import Reference Data](#)^[239]

5.10.2.4 Project Clients

Project clients are the eventual owners of the software system.

Access: [Settings](#) | [Project Types](#) | [People](#) > [Project Clients](#)

Use to:

- Capture client details associated with the current model

Reference:

Field	Usage	See also
Name	Indicates the name of the client.	
Organization	Indicates the name of the organization that employs the client.	
Role(s)	Indicates the role the client plays in the project (for example, Manager, Sponsor).	
Phone 1, Phone 2, Mobile, Fax	Indicates the contact telephone numbers for the client.	
Email	Indicates the email address of the client.	
Notes	Indicates any additional notes on the client.	
Defined Clients	Review clients that have already been defined.	
New	Adds details of further clients.	
Save	Adds a new client to the Defined Client list.	
Delete	Deletes a client record from the Defined Client list.	

Notes:

- You can transport these client definitions between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.10.3 General Types

The General Types dialog enables you to configure:

- Status Types
- Constraint Types
- Constraint Status Types
- Requirement Types

- Scenario Types

Access: [Settings](#) | [Project Types](#) | [General Types](#)

Learn More:

- [Status Types](#) ^[783]
- [Constraint Types](#) ^[784]
- [Constraint Status Types](#) ^[785]
- [Requirement Types](#) ^[785]
- [Scenario Types](#) ^[786]

5.10.3.1 Status Types

You can configure a basic list of status types used in Enterprise Architect. Note that whilst most dialogs use this list, not all do so.

Access: [Settings](#) | [Project Types](#) | [General Types](#) > [Status](#)

Use to:

- Configure Status Types

Reference:

Field	Usage	See also
Status	Indicates the name of the status.	
Description	Indicate a short description for the status.	
Status Type Color	Enables a color to be applied to a Status Type.	
Preview	Provides a visual representation of a Status.	
Restore Default	Restores the Status Type Color to it's default.	
Applies to...	By default, status colors only apply to Requirement, Issue and Change elements. You might decide to also apply these colors to other UML elements, such as Use Cases or Classes. To do this, click on the Applies to... button and select the checkbox against each required element type in the Applied Status Colors list.	Requirement ^[1154] , Issue ^[1719] Change ^[1721]
New	Creates a new status.	
Save	Saves the status, adds the status to the status list.	
Delete	Deletes a status that has been highlighted.	
Type List	This list details the current status types and descriptions for each status.	

Notes:

- You can transport the status types (and the colors assigned to status types) between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export**

menu

- To ensure status colors display on your diagrams, open the Options dialog at the Objects page and select the Show status colors on diagrams checkbox
- Requirement, Feature, Issue and Change elements have a status color compartment, but other elements do not. The status color for these elements is applied to the element shadow. Therefore, on the Options dialog Diagram Appearance page you must also select the Element Shadows on checkbox

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]
- [Object Appearance](#)^[432]
- [Diagram Appearance](#)^[427]

5.10.3.2 Constraint Types

The **Constraint** tab of the **General Types** dialog enables you to define constraints. These are picked up in a variety of places where constraints might fall into more categories than the basic (default) *Pre-*, *Post-* and *Invariant* conditions.

Access: **Settings | Project Types | General Types > Constraint**

Use to:

- Allows for definition of constraints

Reference:

Field	Usage	See also
Constraint	Indicates the name of the constraint.	
Description	Indicates a brief description of the constraint.	
Note	Indicates any additional information required.	
Defined Constraint Type	A list of defined constraint types.	
New	Add a new constraint.	
Save	Saves the constraint and adds it to the Defined Constraint Type list.	
Delete	Removes a constraint from the Defined Constraint Type list.	

Notes:

- You can transport these constraints between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.10.3.3 Constraint Status Types

You can configure the basic list of *constraint status types* used in Enterprise Architect.

Access: [Settings](#) | [Project Types](#) | [General Types](#) > [Constraint Status](#)

Use to:

- Allows for configuration of basic constraint status types

Reference:

Field	Usage	See also
Status	Indicates the name of the constraint status.	
Status List	A list of defined constraint types.	
New	Add a new constraint status.	
Save	Saves the constraint status and adds it to the Status list.	
Delete	Removes a constraint status from the Status list.	

Notes:

- You can transport these constraint status types between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.10.3.4 Requirement Types

The **Requirement** tab of the **General Types** dialog enables you to specify the generic set of requirement types that can be entered into the requirements sections of dialogs. This helps to maintain a single set of typed requirements.

Access: [Settings](#) | [Project Types](#) | [General Types](#) > [Requirement](#)

Use to:

- Specify the generic set of requirement types

Reference:

Field	Usage	See also
Requirement	Indicates the name of the requirement.	
Description	Indicates a short description of the requirement.	

Weight	Indicates a weighting to apply to the requirement type.	
Defined Requirement Types	Displays all of the pre-defined and saved requirement types.	
New	Add a new requirement.	
Save	Saves the requirement details and adds it to the Defined Requirement Types list.	
Delete	Deletes a requirement from the Defined Requirement Types list.	

Notes:

- You can transport these requirement types between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.10.3.5 Scenario Types

A drop-down list of scenario types is available in the Scenario tab of an element Properties dialog, with the standard types *Basic Path*, *Exception* and *Alternate Flow*. You can set additional scenario types using the Scenario tab the General Types dialog.

Access: **Settings | Project Types | General Types > Scenario**

Use to:

- Add additional Scenario Types

Reference:

Field	Usage	See also
Scenario Type	Indicates the name of the scenario.	
Description	Indicates a short description of the scenario.	
Weight	Indicates a weighting to apply to the scenario type.	
Note	Indicates any additional information on the scenario type.	
Defined Scenario Types	Displays all of the pre-defined and saved scenario types.	
New	Add a new scenario.	
Save	Saves the scenario details and adds it to the Defined Scenario Types list.	
Delete	Deletes a scenario from the Defined Scenario Types list.	

Notes:

- You can transport these scenario types between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu

Learn More:

- [Scenarios](#) ^[667]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

5.10.3.6 Metrics and Estimation

TCF values, EFC values and Default Hour Rate for a project are controlled from the **Estimation Factors** dialog.

Risk, metric and effort types for a project are controlled from the **Project Indicators** dialog.

For further information on these see the [Project Management](#) ^[346] and [Resource Management](#) ^[348] topics, or specifically:

- [Technical Complexity Factors](#) ^[401]
- [Environment Complexity Factors](#) ^[402]
- [Default Hours](#) ^[404]
- [Effort Types](#) ^[353]
- [Metric Types](#) ^[354]
- [Risk Types](#) ^[355]

5.10.4 Maintenance

Access: **Settings | Project Types | Maintenance**

Use to:

- Set the base Problem Types that are handled

Learn More:

- [Testing Types](#) ^[788]

5.10.4.1 Problem Types

NOT CURRENTLY USED

For the maintenance and change control screens, you can use the **Maintenance** dialog to set the base *Problem Types* that are handled. Examples are hardware-related issues, performance problems, software bugs and network problems.

Access: **Settings | Project Types | Maintenance > Problem Types**

Use to:

- Set the base Problem Types that are handled

Reference:

Field	Usage	See also
Problem Type	Indicates the name of the problem type.	
Description	Indicates a short description of the problem type.	
Weight	Indicates a weighting to apply to the problem type.	
Note	Indicates any additional information on the problem type.	
Defined Types	Displays all of the pre-defined and saved problem types.	
New	Add a new problem type.	
Save	Saves the scenario details and adds it to the Defined Types list.	
Delete	Deletes a scenario from the Defined Types list.	

Notes:

- You can transport these problem types between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu
- You transport the problem types together with test types as a *Maintenance Types* file

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.10.4.2 Testing Types

Use the **Test Types** tab of the **Maintenance** dialog to add testing types to the basic set that comes with Enterprise Architect. Typical test types are load tests, performance tests and function tests.

Access: **Settings | Project Types | Maintenance > Test Types**

Use to:

- Add Testing Types

Reference:

Field	Usage	See also
Test Type	Type the name of the test type.	
Description	Type a short description of the test type.	
Weight	Type a weighting to apply to the test type.	
Note	Type any additional information on the test type.	
Defined Types	Displays all of the pre-defined and saved test types.	

New	Add a new test type.	
Save	Saves the scenario details and adds it to the Defined Types list.	
Delete	Deletes a scenario from the Defined Types list.	

Notes:

- You can transport these test types between models, using the Export Reference Data and Import Reference Data options on the **Project | Model Import/Export** menu
- You can either export the test types together with the default problem types, as a *Maintenance Types* file, or separately as a *Test Types* file

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

5.10.5 Resources

The **Resources** window displays a tree of MDG Technologies, Templates, Documents, UML Profiles and Patterns, commonly-used model elements and Matrix profiles. This view provides useful shortcuts and re-use functions that you can use to add stock elements to the current model, and patterns and elements for additional information.

Access: [View | More Project Tools > Project Resources](#)

Use to:

- Add Stock elements to the current model
- Add patterns and elements

Reference:

Field	Usage	See also
MDG Technologies	Provides a convenient way to insert complex new elements and features without having to retype or reconfigure each element.	MDG Technologies ^[1031]
Templates	Provides a range of templates for creating HTML (web) reports, RTF reports in either the legacy report generator or the extended RTF report generator, linked documents and MDG Technology reports; you can create, edit, copy and delete your own templates, and view and copy system-supplied or technology-supplied templates using context menu options.	HTML (web) ^[1804] , RTF ^[1726] , legacy report generator ^[1787] , Extended RTF report generator ^[1727] , linked documents ^[729]
Documents	Provides a shortcut to saved RTF documents.	
Matrix Profiles	Provides quick access to saved Relationship Matrix profiles; double-click on a profile to load the matrix with the saved settings and source-target packages.	
Favorites	Provides a shortcut to elements that you configure as a shortcut	
Stylesheets	Enables you to import XSL Style sheets, which are then available in	

Field	Usage	See also
	the drop-down list on the XML Export dialog.	
UML Profiles	Provides a convenient way to insert complex new elements and features without having to retype or reconfigure each element.	UML Profiles ^[1026]
UML Patterns	Provides a convenient way to insert complex new elements and features without having to retype or reconfigure each element.	UML Patterns ^[1027]

Notes:

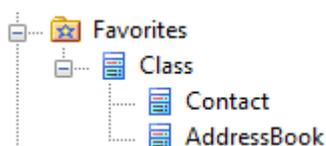
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Configure Resources permission to maintain Resources window items
- From release 7.5 of Enterprise Architect, a method of importing MDG Technologies into the **Resources** window is available but *Not Recommended*. However, you might previously have imported Technologies into the **Resources** window, and these are still available until you specifically delete them (right-click on the Technology and select the **Delete Technology** context menu option). The preferred method of making MDG Technologies available to your users is to import them into the % APPDATA% folder.
- You can also synchronize the Tagged Values and constraints for any elements created from a profile element in the **Resources** window
- You To add a document to the shortcut list, select the **Project | Documentation | Rich Text Format (RTF) Report** menu option. Once you have defined your document click on the **Resource Document** button and type in a name. The document name then displays in the **Resources** window. By right-clicking on the document name you can regenerate documents individually or as a batch, or open them directly from Enterprise Architect
- If you select a style sheet on export, Enterprise Architect applies that style sheet to the XML generated before saving to file. This makes it convenient to generate other forms of output from the base XML content. Combined with UML Profiles, this is a powerful means of extending Enterprise Architect to generate almost any content required

Learn More:

- [Permission List](#) ^[205]
- [importing MDG Technologies](#) ^[1035]
- [Synchronize Tagged Values and Constraints](#) ^[1029]
- [Resource Documents](#) ^[1766]

5.10.5.1 Favorites

The **Resources** window contains a *Favorites* folder. Here you can hyperlink to any UML element from the model as a whole, and conveniently drag and drop instances or links to this element into other diagrams. This is particularly useful where certain elements - such as the list of Actors in a system - are re-used again and again, and switching to the *Actors* folder is not convenient. In cases like this, using the Favorites folder makes managing and creating your model much easier.

**Topics:**

Topic	Detail	See also
Add to the Favorites Folder	To add an element to the Favorites folder: <ul style="list-style-type: none">• In a diagram, right-click on the element to add• From the context menu select the Find Add to Favorites option• Switch back to the Resources window and check the Favorites folder; the new element should be listed in its category within the favorites	
Delete from the Favorites Folder	To delete a favorite: <ul style="list-style-type: none">• Right-click on it within the Favorites folder in the Resources window• Select Delete Favorite from the context menu• Confirm the action by clicking on the Yes button	
View Properties of a Favorite	To view a favorite's properties from the Favorites folder: <ul style="list-style-type: none">• Select and right-click on the favorite in the Resources window• Select Element Properties from the context menu	

Part



6 Standard UML Models



This topic provides an introduction to Enterprise Architect's diagrams, elements and connectors, and its modeling process.

It also illustrates its alignment to the Unified Modeling Language (UML) 2.3, an open modeling standard, defined and maintained by the Object Management Group.

Topic	Detail	See also
The Unified Modeling Language (UML)	<p>The UML standard defines notations and rules for specifying business and software systems; the notation supplies a rich set of graphic elements for modeling object oriented systems, and the rules state how those elements can be connected and used</p> <p>UML is not a tool for creating software systems; instead, it is a visual language for communicating, modeling, specifying and defining systems</p> <p>UML is not a prescriptive process for modeling software systems; it does not supply a method or process, simply the language. You can therefore use UML in a variety of ways to specify and develop your software engineering project</p> <p>This language is designed to be flexible, extendable and comprehensive, yet generic enough to serve as a foundation for all system modeling requirements. With its specification, there is a wide range of elements characterized by the kinds of diagrams they serve, and the attributes they provide. All can be further specified by using stereotypes, Tagged Values and profiles.</p> <p>Enterprise Architect supports many different kinds of UML elements (as well as some custom extensions); together with the connectors between elements, these form the basis of the model</p>	<p>UML Diagrams ^[795]</p> <p>UML Elements ^[864]</p> <p>UML Connectors ^[966]</p> <p>UML Stereotypes ^[1016]</p> <p>UML Profiles ^[1026]</p> <p>Tagged Values ^[762]</p>
Wide Range of Applications	<p>Although initially conceived as a language for software development, UML can be used to model a wide range of real world domains and processes (in business, science, industry, education and elsewhere), organizational hierarchies, deployment maps and much more</p> <p>Enterprise Architect also provides additional custom diagrams and elements, to address further modeling interests.</p>	<p>Modeling Basics ^[515]</p>
Extending UML for New Domains	<p>Using UML Profiles, UML Patterns, Grammars, Data Types, Constraints, MDG Technologies and other extensions, UML and Enterprise Architect can be tailored to address a particular modeling domain not explicitly covered in the original UML specification</p>	<p>UML Profiles ^[1026]</p> <p>UML Patterns ^[1021]</p> <p>MDG Technologies ^[1031]</p>

Topic	Detail	See also
	Enterprise Architect makes extending UML simple and straightforward and, best of all, the extension mechanism is still part of the UML Specification.	
Recommended Reading	In addition to the UML Specification available from the OMG, two books that provide excellent introductions to UML are: <ul style="list-style-type: none"><li data-bbox="539 504 1118 616">• <i>Schaum's Outlines: UML</i> by Bennet, Skelton and Lunn Published by McGraw Hill. ISBN 0-07-709673-8<li data-bbox="539 627 1118 710">• <i>Developing Software with UML</i> by Bern Oestereich Published by Addison Wesley. ISBN 0-201-36826-5	

Learn More:

- [OMG website](#)
- Diagram [Toolbox](#)^[546] descriptions
- The *EASample.eap* project supplied with Enterprise Architect
- Online [UML Tutorial](#) (parts 1 and 2) and [UML 2.0 Tutorial](#)

6.1 UML Diagrams

A UML diagram is a representation of the components or elements of a system or process model and, depending on the type of diagram, how those elements are connected or how they interact from a particular perspective. For example, how and why an object changes state, or how requirements are realized by the process or a system.

Topics:

Topic	Detail	See also
Working with Diagrams	Diagrams are developed in the main workspace, in which you create and connect model elements You create diagrams by right-clicking a package and selecting the New Diagram context menu option, or load them by double-clicking on their diagram icon in the Project Browser	Diagram Tasks ^[567]
Types of Diagram	There are two major groupings of UML diagrams: <ul style="list-style-type: none"> • Structural Diagrams which depict the structural elements composing a system or function, reflecting the static relationships of a structure, or run-time architectures. • Behavioral Diagrams which show a dynamic view of the model, depicting the behavioral features of a system or business process. 	Structural Diagrams ^[796] Behavioral Diagrams ^[817]
Extended Diagrams	Enterprise Architect provides a set of additional diagram types that extend the core UML diagrams for domain-specific models	Extended Diagrams ^[1187]
Custom Diagrams	Enterprise Architect also supports diagram types specific to MDG Technologies, including integrated technologies	Specialized UML Models ^[1178]

6.2 UML Structural Models

Structural diagrams depict the structural elements composing a system or function. These diagrams reflect the static relationships of a structure, such as Class or Package diagrams, or run-time architectures such as Object or Composite Structure diagrams.

Structural diagrams include the following diagram types:

Diagram Type	Detail	See also
Class Diagrams	Class diagrams capture the logical structure of the system, the Classes and objects that make up the model, describing what exists and what attributes and behavior it has.	Class diagrams ^[798]
Composite Structure Diagrams	Composite Structure diagrams reflect the internal collaboration of Classes, Interfaces and Components (and their properties) to describe a functionality.	Composite Structure diagrams ^[801]
Component Diagrams	Component diagrams illustrate the pieces of software, embedded controllers and such that make up a system, and their organization and dependencies.	Component diagrams ^[807]
Deployment Diagrams	Deployment diagrams show how and where the system is to be deployed; that is, its execution architecture.	Deployment diagrams ^[804]
Object Diagrams	Object diagrams depict object instances of Classes and their relationships at a point in time.	Object diagrams ^[799]
Package Diagrams	Package diagrams depict the organization of model elements into packages and the dependencies amongst them.	Package diagrams ^[796]
Profile Diagrams	Profile Diagrams are those created in a «profile» package, to extend UML elements, connectors and components.	Profile Diagrams ^[809]

6.2.1 Package Diagram

Package diagrams depict the organization of model elements into packages and the dependencies amongst them, including package imports and package extensions. They also provide a visualization of the corresponding namespaces.

Use To:

- depict the organization of model elements into packages
- show dependencies between packages
- visualize namespaces

Example Diagram: [Example Package Diagram](#) ^[797]

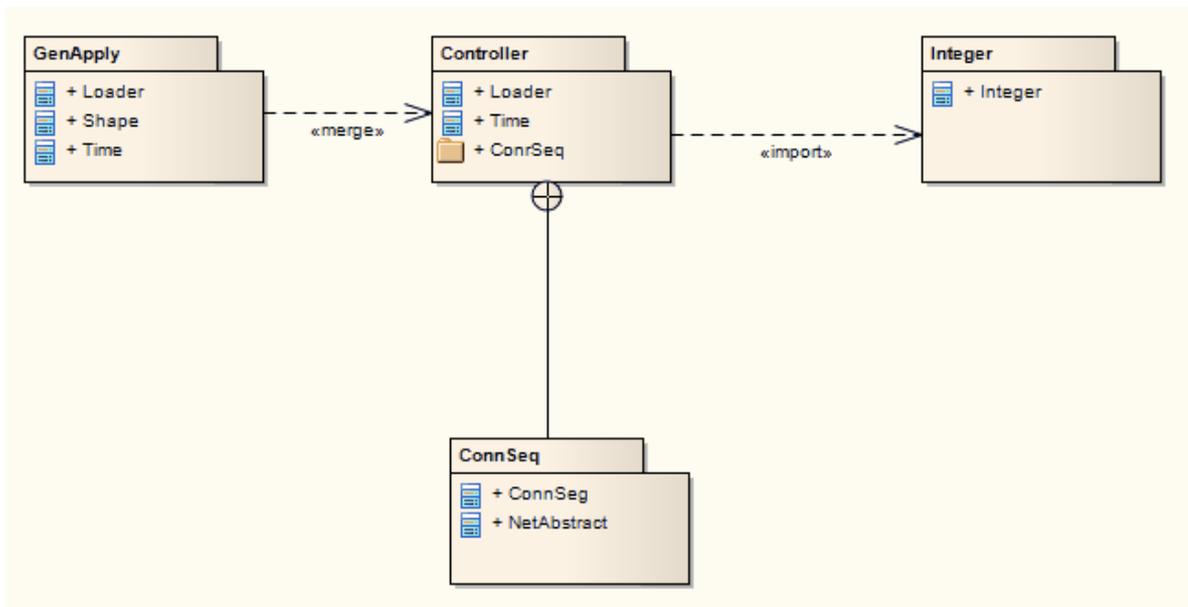
Tools:

Select Package diagram elements and connectors from the Class pages of the Toolbox.

Package Diagram Elements	Package Diagram Connectors
Package	Associate
Class	Generalize
Interface	Compose
Data Type	Aggregate
Enumeration	Association Class
Primitive	Assembly
Table	Realize
Signal	Nesting
Association	Package Merge
	Package Import

6.2.1.1 Example Package Diagram

The following example demonstrates a basic Package diagram.



Item	Description	See Also
Nesting connector between ConnSeq and Controller	The nesting connector between ConnSeq and Controller reflects what the package contents reveal. Package contents can be listed by clicking on the diagram background to display	Properties ^[569] dialog

Item	Description	See Also
	the diagram's Properties dialog, selecting the Elements tab and selecting the Package Contents checkbox.	
«import» connector	The «import» connector indicates that the elements within the target Integer package, which in this example is the single Class Integer, are imported into the package Controller. The Controller's namespace gains access to the Integer Class; the Integer namespace is not affected.	
«merge» connector	The «merge» connector indicates that the package Controller's elements are imported into GenApply, including Controller's nested and imported contents. If an element already exists within GenApply, such as Loader and Time, these elements' definitions are expanded by those included in the package Controller. All elements added or updated by the merge are noted by a generalization relationship back to that package.	

6.2.2 Class Diagram

The Class diagram captures the logical structure of the system - the Classes - and things that make up the model. It is a static model, describing what exists and what attributes and behavior it has, rather than how something is done.

Use To:

Illustrate relationships between Classes and Interfaces. Generalizations, Aggregations and Associations are all valuable in reflecting inheritance, composition or usage, and connections, respectively.

Example Diagram: [Example Class Diagram](#) ⁷⁹⁹

Tools:

Select Class diagram elements and connectors from the Class pages of the Toolbox.

Class Diagram Elements	Class Diagram Connectors
 Package	 Associate
 Class	 Generalize
 Interface	 Compose
 Data Type	 Aggregate
 Enumeration	 Association Class
 Primitive	 Assembly
 Table	 Realize
 Signal	 Nesting

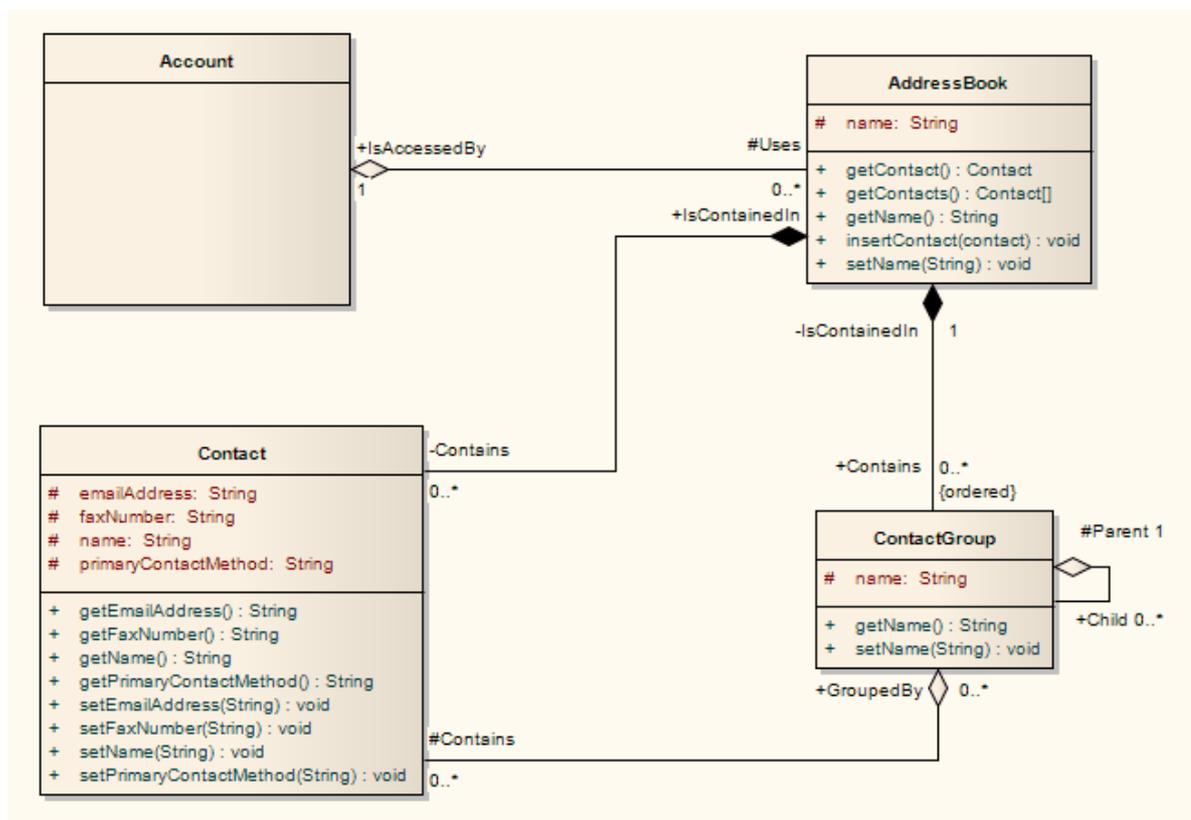
Class Diagram Elements	Class Diagram Connectors
 Association	 Package Merge
	 Package Import

Learn More:

- [Active Classes](#) ^[943]
- [Parameterized \(template\) Classes](#) ^[943]
- [Stereotyped Business Interaction Elements](#) ^[1191]

6.2.2.1 Example Class Diagram

There are two forms of the [Aggregation](#) ^[968] relationship in the following diagram. The pale form indicates that the Class *Account* uses *AddressBook*, but does not necessarily contain *AddressBook*. The dark *Composite* Aggregation form indicates ownership or containment by the target Classes (at the diamond end) of the source Classes.



6.2.3 Object Diagram

An Object diagram is closely related to a Class diagram, with the distinction that it depicts object instances of Classes and their relationships at a point in time. This might appear similar to a Composite Structure diagram, which also models run-time behavior; the difference is that Object diagrams exemplify the static Class diagrams, whereas Composite Structure diagrams reflect run-time architectures different from their

static counterparts. Object diagrams do not reveal architectures varying from their corresponding Class diagrams, but reflect multiplicity and the roles instantiated Classes could serve.

Use To:

They are useful in understanding a complex Class diagram, by creating different cases in which the relationships and Classes are applied. An Object diagram can also be a kind of Communication diagram, which also models the connections between objects, but additionally sequences events along each path.

Example Diagram: [Example Object Diagram](#) ^[800]

Tools:

Select Object diagram elements and connectors from the Object pages of the Toolbox.

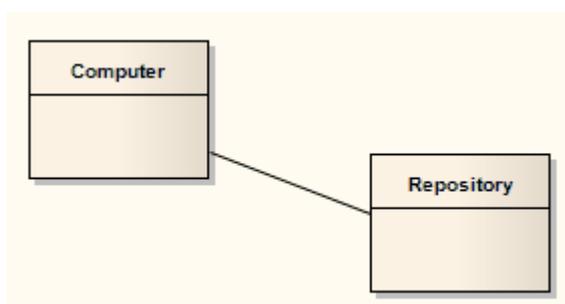
Object Diagram Elements	Object Diagram Connectors
 Actor	 Information Flow
 Object	 Associate
 Collaboration	 Dependency
 Collaboration Use	
 Information Item	
 Boundary	
 Control	
 Entity	

See Also:

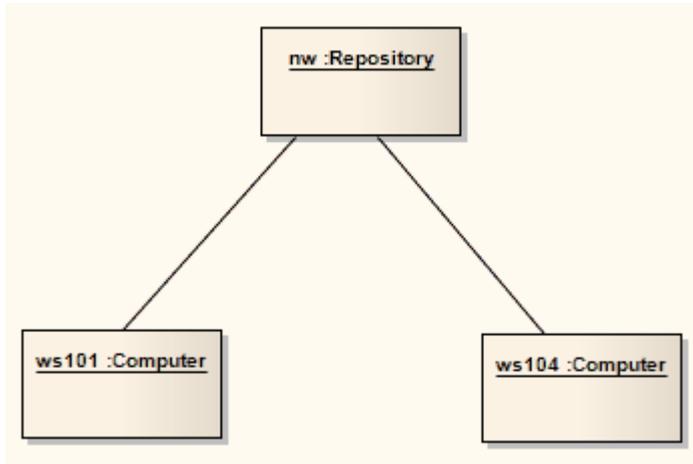
- [Business Modeling Objects](#) ^[1191]
- [Communication diagram](#) ^[859]

6.2.3.1 Example Object Diagram

The following example first shows a simple Class diagram, with two Class elements connected.



The Classes above are instantiated below as Objects in an Object diagram. There are two instances of **Computer** in this model, which can prove useful for considering the relationships and interactions Classes play in practice, as Objects.



6.2.4 Composite Structure Diagram

A Composite Structure diagram reflects the internal collaboration of Classes, Interfaces or Components (and their Properties) to describe a functionality. Composite Structure diagrams are similar to Class diagrams, except that they model a specific usage of the structure. Class diagrams model a static view of Class structures, including their attributes and behaviors.

Use To:

A Composite Structure diagram is used to express run-time architectures, usage patterns and the participating elements' relationships, which might not be reflected by static diagrams.

In a Composite Structure diagram, Classes are accessed as Parts or run-time instances fulfilling a particular role. These Parts can have multiplicity, if the role filled by the Class requires multiple instances. Ports defined by a Part's Class should be represented in the composite structure, maintaining that all connecting Parts provide the required interfaces specified by the Port. There is extensive flexibility, and an ensuing complexity, that come with modeling composite structures. To optimize your modeling, consider building Collaborations to represent reusable patterns responding to your design issues.

Example Diagram: [Example Composite Structure Diagram](#) ^[802]

Tools:

Select Composite Structure diagram elements and connectors from the Composite pages of the Toolbox.

Composite Structure Diagram Elements	Composite Structure Diagram Connectors
Class	Connector
Interface	Assembly
Part	Role Binding
Port	Represents

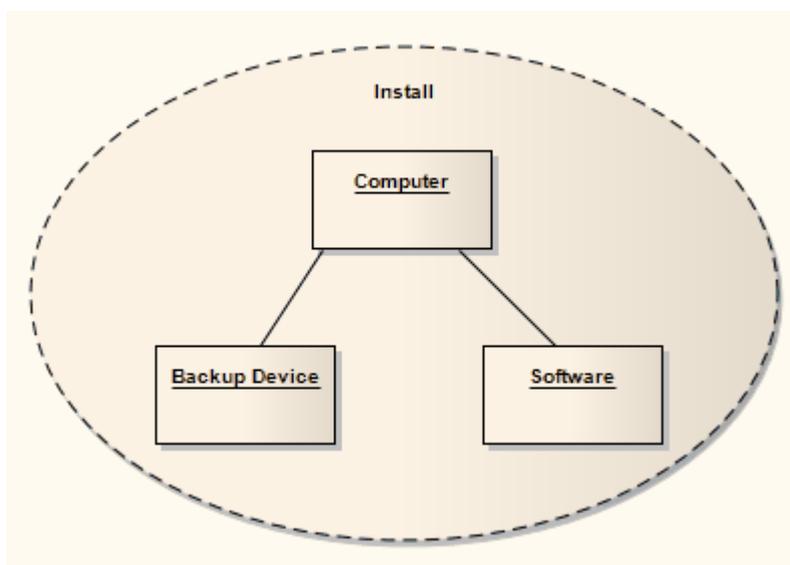
Composite Structure Diagram Elements	Composite Structure Diagram Connectors
 Collaboration	 Occurrence
 Collaboration Use	 Delegate
 Expose Interface	

Learn More:

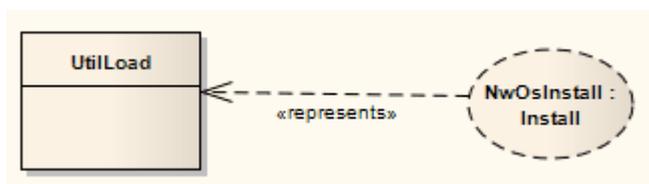
- [Business Use Case Realization](#)^[119]

6.2.4.1 Example Composite Structure Diagram

The following diagram shows a Collaboration used in Composite Structure diagrams to model common patterns. This particular example shows a relationship for performing an installation.



The following diagram uses the Install Collaboration in a Collaboration Use, and applies it to the UtilLoad Class via a «represents» relationship. This indicates that the classifier UtilLoad uses the collaboration pattern within its implementation.



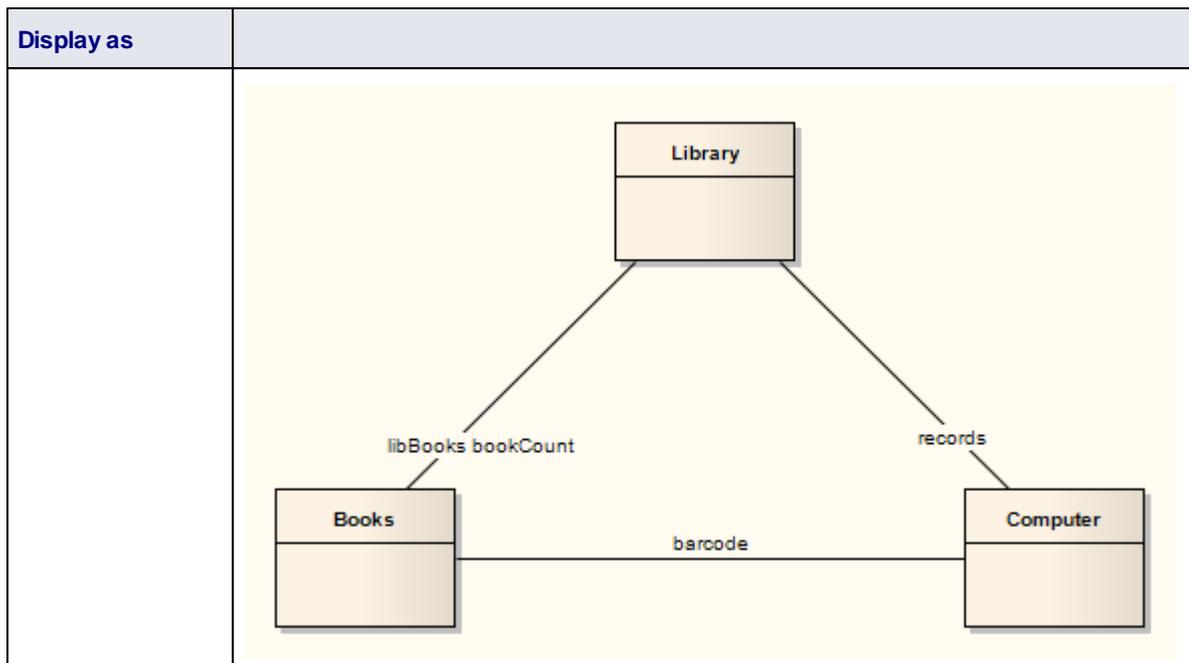
See Also:

- [Collaboration Use](#)^[946]

6.2.4.2 Properties

A *property* is a nested structure within a classifier, which is usually a Class or an Interface, on a Composite Structure diagram. The contained structure reflects instances and relationships reflected within the containing classifier. Properties can have multiplicity.

<p>Display as</p>	
<p>Parts</p>	<div data-bbox="517 546 1401 987" data-label="Diagram"> <pre> classDiagram class Library class libBooks["libBooks :Books [bookCount]"] class records["records :Computer"] Library -- libBooks Library -- records libBooks -- records : barcode </pre> </div> <p>There are two Parts, libBooks and records, which are instances corresponding to the Classes Books and Computer respectively. After dragging Parts from the Toolbox out to the workspace, right-click on a Part and select the Advanced Set Property Type context menu option to connect to a classifier.</p> <p>If Parts disappear when dragged onto the Class, adjust the Z-order of the Class (right-click on it and select the Z-Order context menu option).</p> <p>The relationship between the two Parts is indicated by the connector, reflecting that communication between the Parts is via the barcode. This contained structure and its Parts are properties owned by the Library Class. To indicate a property that is not owned by composition to the containing classifier, use a box symbol with a dashed outline, indicating association. To do this, right-click on the Part and select the Properties context menu option, then select the Advanced page of the Properties dialog and set the IsReference option to true.</p>
<p>Association Roles</p>	<p>Properties can also be reflected using a normal composite structure (without containing it in a Class), with the appropriate connectors, parts and relationships indicated through connections to the Class. This alternative representation is shown in the following diagram. However, this depiction fails to express the ownership immediately reflected by containing properties within a classifier.</p>



6.2.5 Deployment Diagram

A Deployment diagram shows how and where the system is to be deployed; that is, its execution architecture. Hardware devices, processors and software execution environments (system Artifacts) are reflected as Nodes, and the internal construction can be depicted by embedding or nesting Nodes. Deployment relationships indicate the deployment of Artifacts, and Manifest relationships reveal the physical implementation of components. As Artifacts are allocated to Nodes to model the system's deployment, the allocation is guided by the use of deployment specifications.

Example Diagram: [Example Deployment Diagram](#)^[806]

Tools:

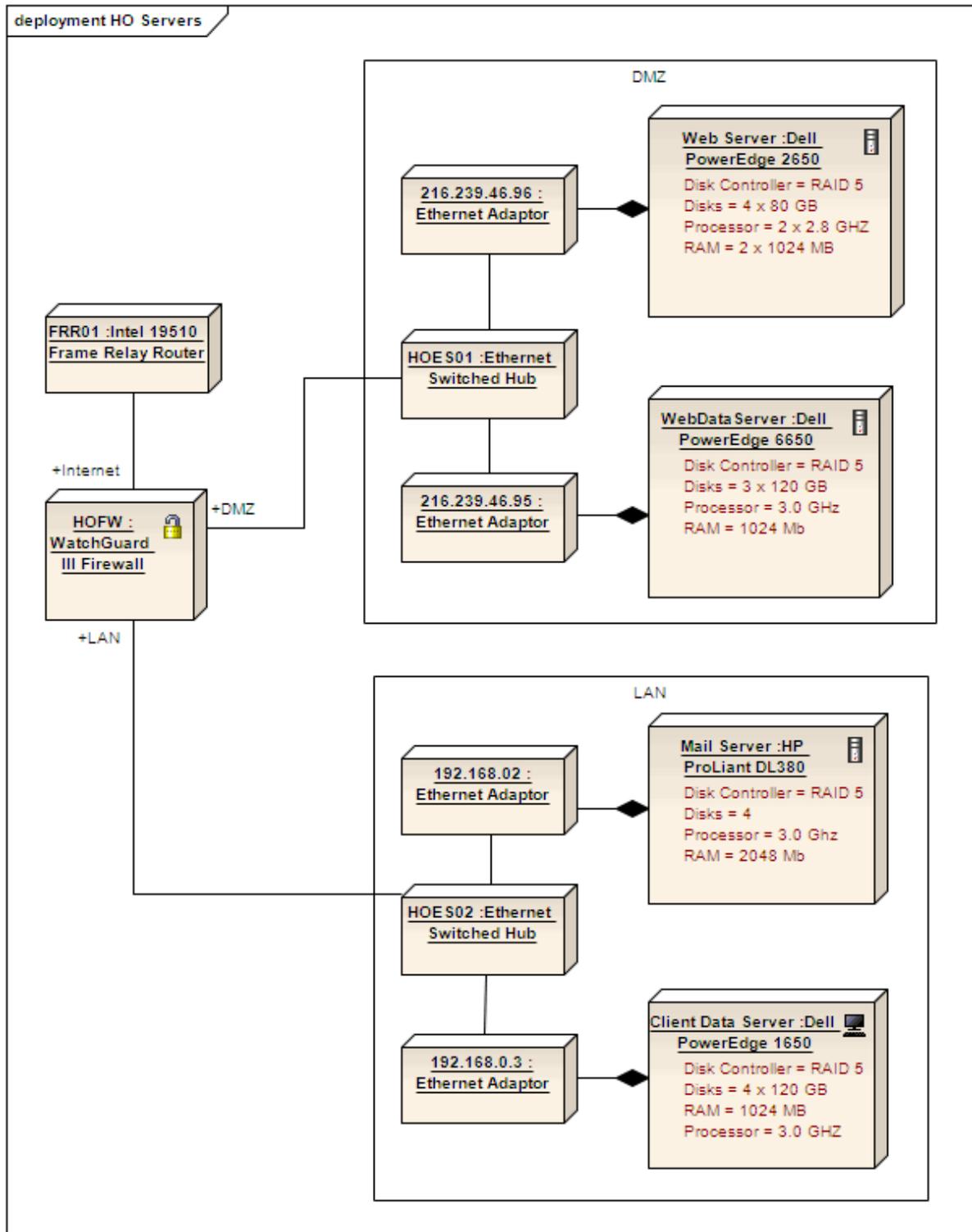
Select Deployment diagram elements and connectors from the Deployment pages of the Toolbox.

Deployment Diagram Elements	Deployment Diagram Connectors
 Node	 Associate
 Device	 Communication Path
 Execution Environment	 Association Class
 Component	 Generalize
 Interface	 Realize
 Artifact	 Deployment
 Document Artifact	 Manifest

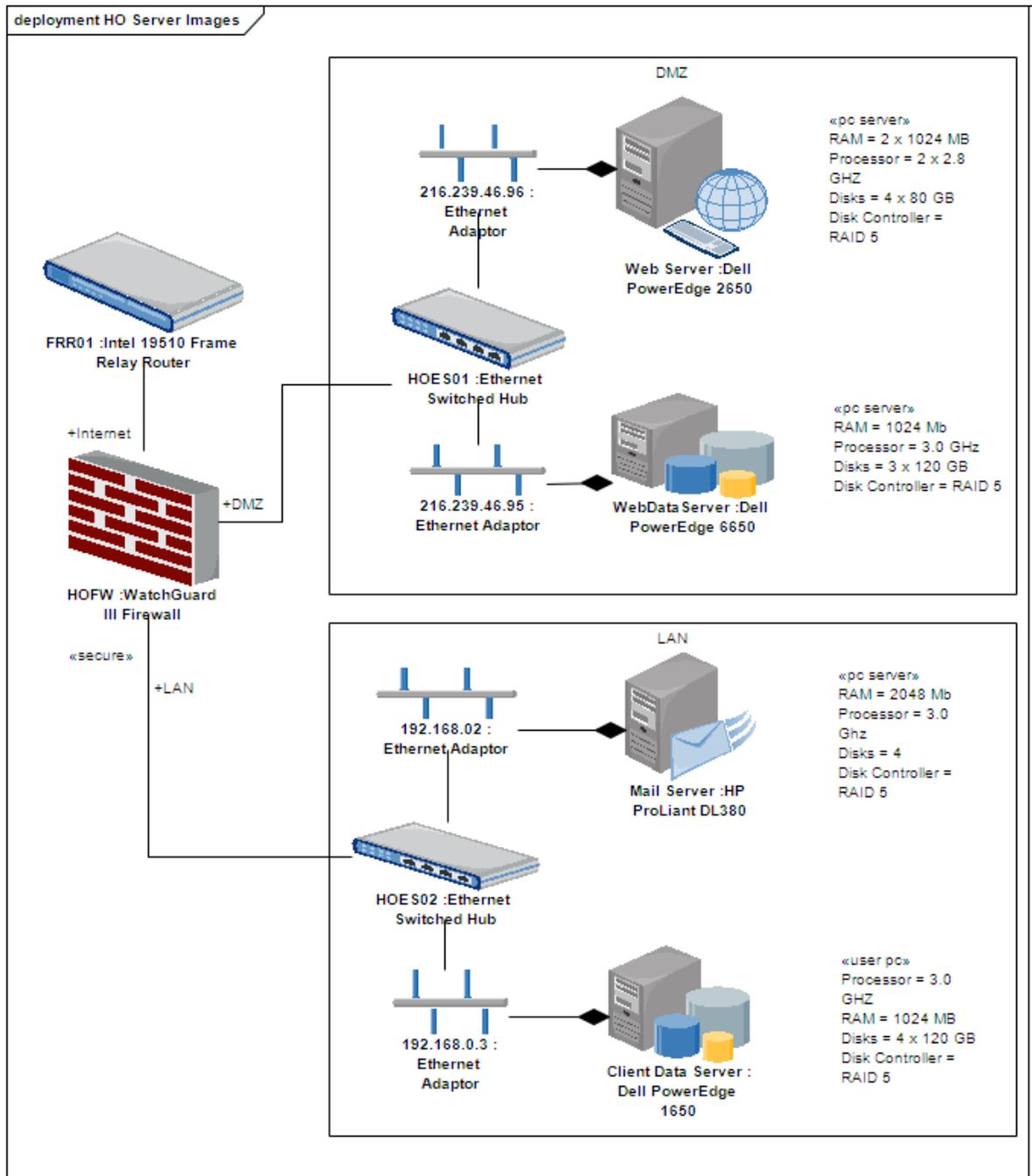
Deployment Diagram Elements	Deployment Diagram Connectors
 Deployment Specification	 Nesting
 Package	

6.2.5.1 Example Deployment Diagram

A simple Deployment diagram is shown below, representing the arrangement of servers at a head office. The servers are represented by Nodes linked by either simple or aggregate Association relationships.



Deployment diagrams are ideal for using alternative images for the objects that the elements represent. Such images can be substituted for the elements in the above diagram, as shown below:



See Also:

- [Alternative Images](#) ^[593]

6.2.6 Component Diagram

A Component diagram illustrates the pieces of software, embedded controllers and such that make up a system, and their organization and dependencies. A Component diagram has a higher level of abstraction than a Class diagram; usually a component is implemented by one or more Classes (or Objects) at runtime. They are building blocks, built up so that eventually a component can encompass a large portion of a system.

Example Diagram: [Example Component Diagram](#) ^[808]

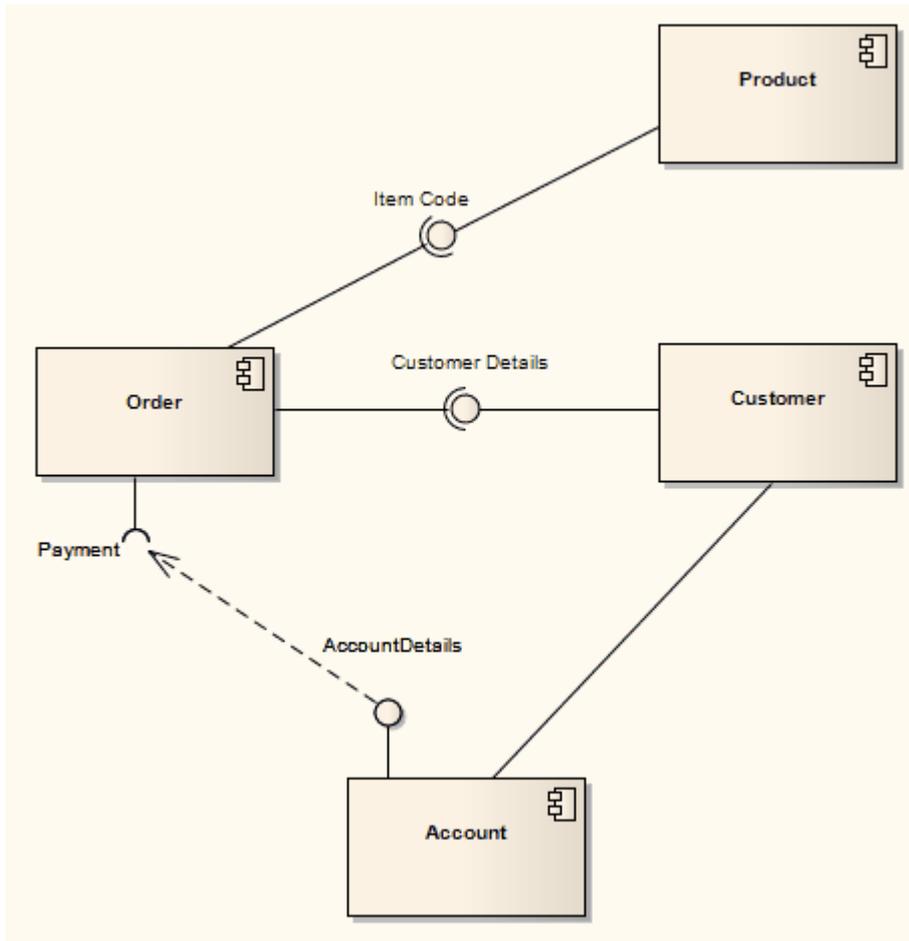
Tools:

Select Component diagram elements and connectors from the Component pages of the Toolbox.

Component Diagram Elements	Component Diagram Connectors
 Package	 Assembly
 Packaging Component	 Delegate
 Component	 Associate
 Class	 Realize
 Interface	 Generalize
 Object	
 Port	
 Expose Interface	
 Artifact	
 Document Artifact	

6.2.6.1 Example Component Diagram

The following diagram demonstrates some components and their inter-relationships. Assembly connectors connect the provided interfaces supplied by Product and Customer to the required interfaces specified by Order. A Dependency relationship maps a customer's associated account details to the required interface Payment, indicated by Order.



Learn More:

- [Assembly](#)^[969]
- [Dependency](#)^[979]

6.2.7 Profile Diagram

A Profile diagram is any diagram created in a «profile» package.

Profiles provide a means of extending the UML. They are based on additional stereotypes and Tagged Values that are applied to UML elements, connectors and their components. A Profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain.

Example Diagram: [Example Profile Diagram](#)^[810]

Tools:

Select the following Profile diagram elements and connectors from the Profile pages of the Toolbox.

Profile Diagram Elements	Profile Diagram Connectors
 Profile	 Extension

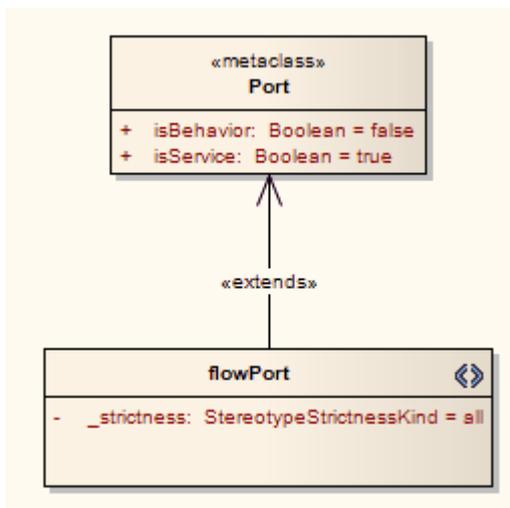
Profile Diagram Elements	Profile Diagram Connectors
 Stereotype	 Generalize
 Metaclass	 Application
 Enumeration	 Tagged Value
	 Redefinition

Learn More:

- [Developing Profiles](#) ^[1039]
- [Create Profiles](#) ^[1047]

6.2.7.1 Example Profile Diagram

A typical unit on a Profile diagram resembles the following illustration:



6.3 UML Behavioral Models

Behavioral diagrams depict the behavioral features of a system or business process. Behavioral diagrams include the following diagram types:

Diagram Type	Detail	See also
Activity Diagrams	Activity diagrams model the behaviors of a system, and the way in which these behaviors are related in an overall flow of the system.	Activity diagrams ^[81]
Use Case Diagrams	Use Case diagrams capture Use Cases and relationships among Actors and the system; they describes the functional requirements of the system, the manner in which external operators interact at the system boundary, and the response of the system.	Use Case diagrams ^[81]
State Machine Diagrams	State Machine diagrams illustrate how an element can move between states, classifying its behavior according to transition triggers and constraining guards.	State Machine diagrams ^[81]
Timing Diagrams	Timing diagrams define the behavior of different objects within a time-scale, providing a visual representation of objects changing state and interacting over time.	Timing diagrams ^[83]
Sequence Diagrams	Sequence diagrams are structured representations of behavior as a series of sequential steps over time. They are used to depict work flow, message passing and how elements in general cooperate over time to achieve a result.	Sequence diagrams ^[84]
Communication Diagrams	Communication diagrams show the interactions between elements at run-time, visualizing inter-object relationships.	Communication diagrams ^[85]
Interaction Overview Diagrams	Interaction Overview diagrams visualize the cooperation between other interaction diagrams (Timing, Sequence, Communication and Interaction Overview diagrams) to illustrate a control flow serving an encompassing purpose.	Interaction Overview diagrams ^[86]

Learn More:

- [Behavioral Modeling](#) ^[70]
- [Code Generation from Behavioral Models](#) ^[149]

6.3.1 Activity Diagram

Activity diagrams are used to model the behaviors of a system, and the way in which these behaviors are related in an overall flow of the system. The logical paths a process follows, based on various conditions, concurrent processing, data access, interruptions and other logical.

Use to:

- Model system behavior
- Model dynamic element interactions

Example: [Example Activity Diagram](#) ^[81]

Tools: Select Activity diagram elements and connectors from the Activity pages of the **Toolbox**. Click on the

following elements and connectors for more information.

Activity Diagram Elements	Activity Diagram Connectors
 Activity	 Control Flow
 Structured Activity	 Object Flow
 Action	 Interrupt Flow
 Partition	
 Object	
 Central Buffer Node	
 Datastore	
 Decision	
 Merge	
 Send	
 Receive	
 Synch	
 Initial	
 Final	
 Flow Final	
 Region	
 Exception	
 Fork/Join	
 Fork/Join	

Notes:

- You can create Analysis diagrams (Simplified Activity), containing the elements most useful for business process modeling, using the New Diagram dialog.

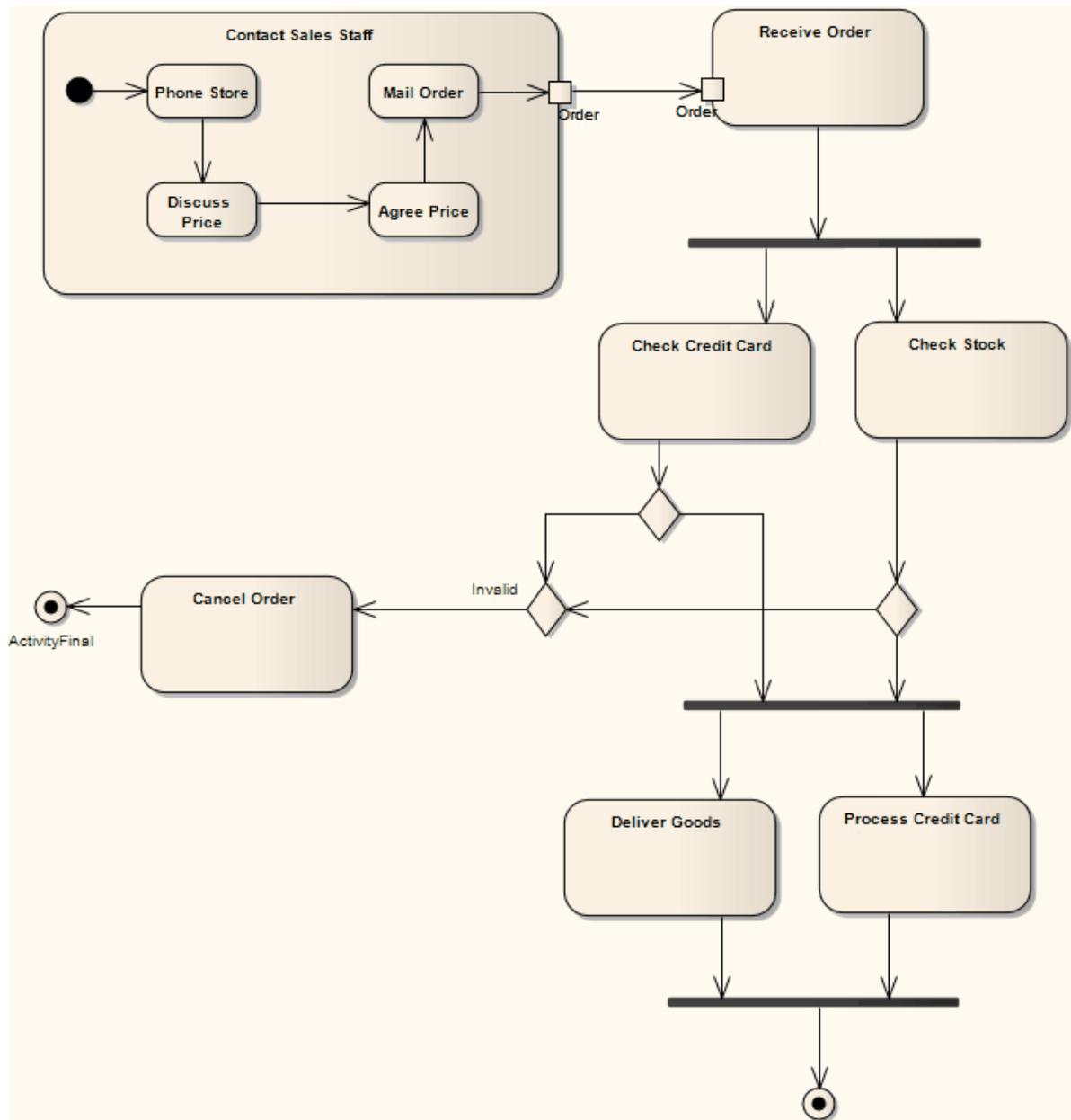
Learn More:

- [Analysis diagrams](#) ^[1187]
- [New Diagram](#) ^[568]

- [Activity pages](#) ^[558]

6.3.1.1 Example Activity Diagram

The following diagram illustrates some of the features of Activity diagrams, including Activities, Actions, Start Nodes, End Nodes and Decision points.



6.3.2 Use Case Diagram

Use Case diagrams capture Use Cases and relationships between Actors and the subject (system).

Use To:

- Describe the functional requirements of the system

- Describe the manner in which outside things (Actors) interact at the system boundary
- Describe the response of the system.

Example Diagram: [Example Use Case Diagram](#) ^[815]

Tools: Select Use Case diagram elements and connectors from the Use Case pages of the **Toolbox**. Click on the following elements and connectors for more information.

Use Case Diagram Elements	Use Case Diagram Connectors
 Actor	 Use
 Use Case	 Associate
 Test Case	 Generalize
 Collaboration	 Include
 Collaboration Use	 Extend
 Boundary	 Realize
 Package	 Invokes
	 Precedes

Notes:

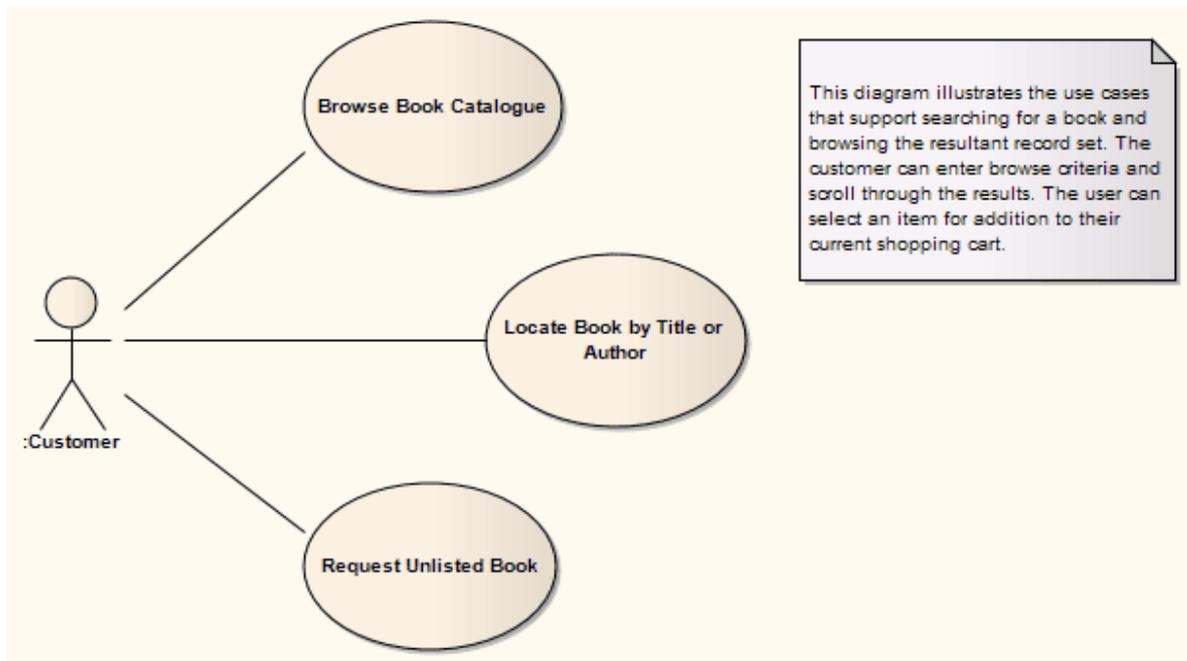
- Invokes and Precedes relationships are defined by the Open Modeling Language (OML). They are stereotyped Dependency relationships; Invokes indicates that Use Case A, at some point, causes Use Case B to happen, whilst Precedes indicates that Use Case C must complete before Use Case D can begin.

Learn More:

- [Use Case Extension Points](#) ^[937]
- [Use Rectangle Notation](#) ^[938]
- [Business Use Case](#) ^[1191]
- [Use Case pages](#) ^[553]

6.3.2.1 Example Use Case Diagram

The following diagram illustrates some features of Use Case diagrams:



6.3.3 State Machine Diagrams

State Machine diagrams illustrate how an element (often a Class) can move between states, classifying its behavior according to transition triggers and constraining guards.

Naming:

- State Machine diagrams were formerly known as State diagrams.
- State Machine representations in UML are based on the Harel State Chart Notation and therefore are sometimes referred to as State Charts.

State Tables:

You can display a State Machine as a diagram (as below) or as a table in one of three relationship formats.

How to:

To select the display format, follow the steps below:

Step	Action	See Also
1	Right-click on the diagram background to display the context menu	
2	Select the Statechart Editor option	
3	Select the appropriate display option: <ul style="list-style-type: none"> • Diagram • Table (State-Next State) 	

Step	Action	See Also
	<ul style="list-style-type: none"> • Table (State-Trigger) • Table (Trigger-State) 	

Example Diagram: [Example State Machine Diagram](#) ^[817]

Tools: Select State Machine diagram elements and connectors from the State pages of the Toolbox. Click on the elements and connectors below for more information.

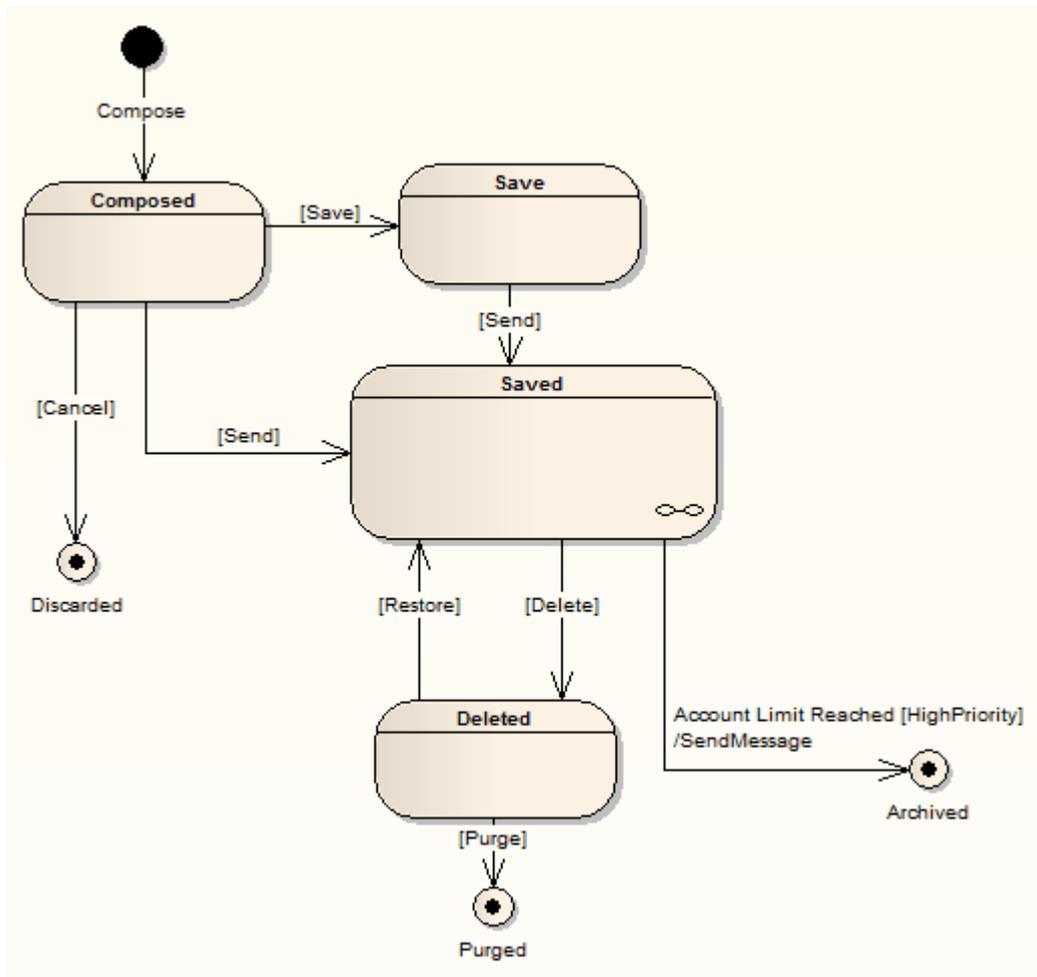
State Machine Diagram Elements	State Machine Diagram Connectors
 State	 Transition
 State Machine	 Object Flow
 Initial	
 Final	
 History	
 Synch	
 Object	
 Choice	
 Junction	
 Entry	
 Exit	
 Terminate	
 Fork/Join	
 Fork/Join	

Learn More:

- [SW Code Generation - State Machine Diagrams](#) ^[1497]
- [State Machine Modeling for HDLS](#) ^[1502]
- [State Machine Table](#) ^[821]
- [State Group](#) ^[558]

6.3.3.1 Example State Machine Diagram

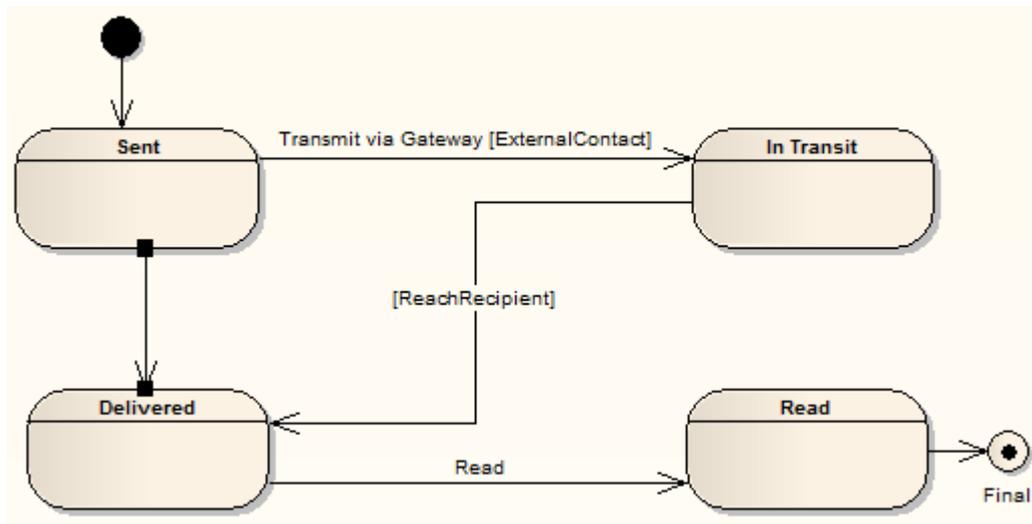
The following diagram illustrates some features of State Machine diagrams.



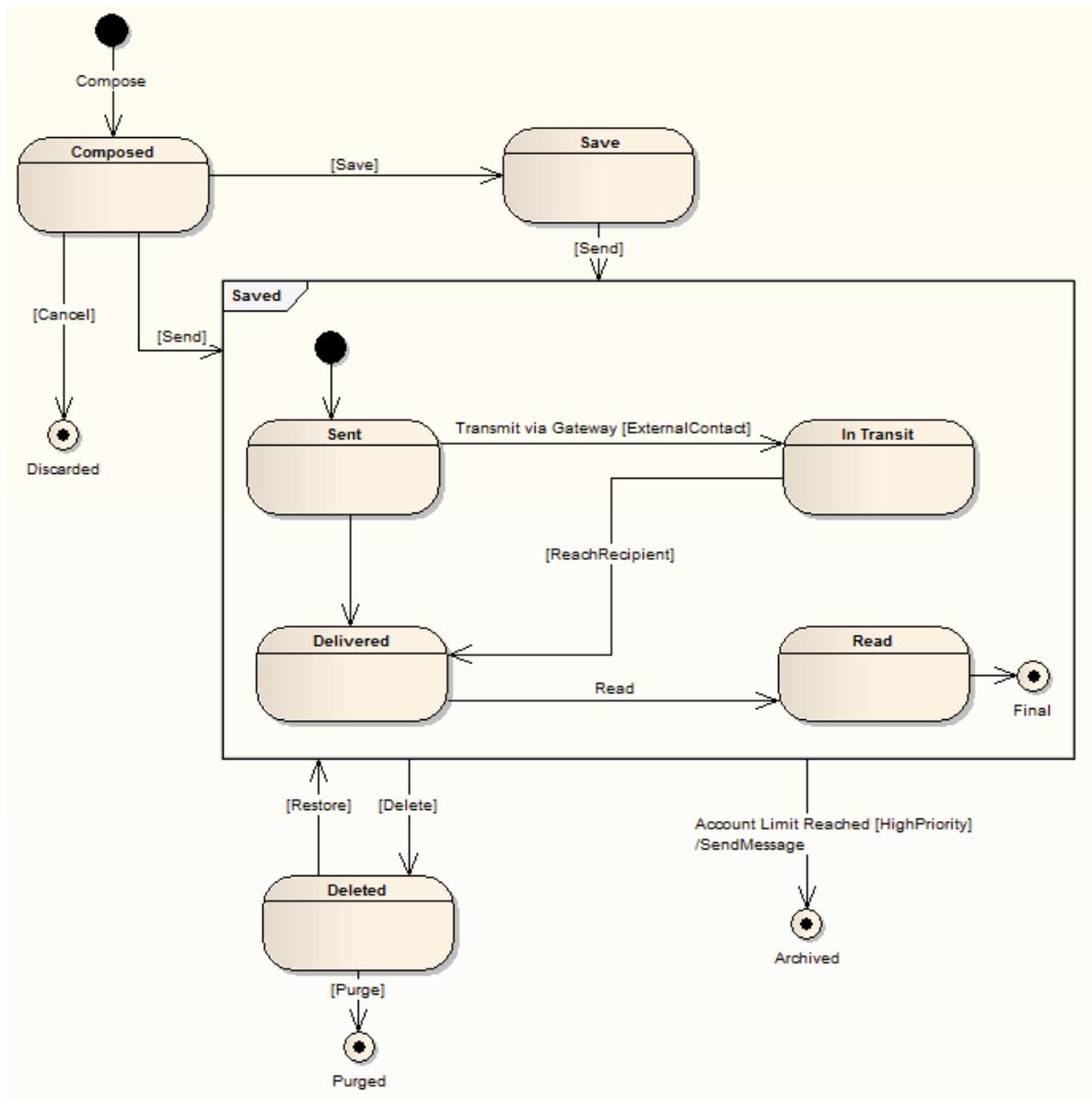
Composite States:

The chain-link symbol in the bottom right corner of the Saved state indicates that it is a Composite state.

You have two options for exposing the contents of a composite State. Firstly, you can double-click on the element to display its child diagram separately, as shown below:



Alternatively, you can right-click on the composite element and select the **Advanced | Show Composite Diagram** context menu option, which displays the child diagram in the context of the parent diagram.

**Notes:**

- State elements can display either with or without a line across them. The line - as shown above - displays when the element has features such as attributes (which could be hidden) or when the Show State Compartment checkbox is selected in the Objects page of the Options dialog. See: [Show State Compartment](#)^[432].

Learn more:

- [Composite States](#)^[919]
- [Pseudo-states](#)^[820]
- [State Elements](#)^[917]
- [Regions](#)^[820]

6.3.3.2 Pseudo-States

Pseudo-states are a UML abstraction for various types of transient vertices used in State Machine diagrams. Pseudo-states are used to express complex transition paths. The following types of pseudo-state are available:

Type	See
 Initial	Initial ^[903]
 Entry	Entry Point ^[891]
 Exit	Exit Point ^[895]
 Choice	Choice ^[879]
 Junction	Junction ^[908]
 History	History ^[902]
 Terminate	Terminate ^[933]
 Final	Final ^[895]
 Fork/Join	Fork ^[900]
 Fork/Join	
 Fork/Join	Join ^[901]
 Fork/Join	

Learn More:

- [State Machine Diagrams](#)^[815]

6.3.3.3 Regions

Regions can be created in Composite States or State Machines on a State Machine diagram. Regions indicate concurrency, such that a single State is active in each region. Multiple transitions can occur from a single event dispatch, so long as similarly triggered transitions are divided by Regions.

How to:

To create a Region in a Composite State or State Machine element, follow the steps below:

Step	Action
1	Right-click on the element, and select the Advanced Define Concurrent Substates context menu option The State Regions dialog displays
2	Create the Regions of a State, which can be named or anonymous

3	Click on the OK button
---	-------------------------------

Learn More:

- [Composite States](#) ^[819]
- [State Machines](#) ^[817]
- [State Machine diagram](#) ^[815]

6.3.4 State Machine Table

A State Machine table is one of two variants of a State Machine (the other is the State Machine diagram). It displays the information of the State Machine in table form, and is a method of specifying the discrete behavior of a finite state-transition system; that is, what state the State Machine moves to and the conditions under which the transition takes place.

You can display the state transition as one of two different relationships:

Display Type	Description	See Also
State - Trigger	The rows indicate the current states and the columns indicate trigger events (or the other way around if you prefer, in a Trigger - State format). The cell at the intersection of a row and column identifies the target state in the transition if the trigger occurs, and the condition (or guard) of the transition.	Example State-Trigger Table ^[829]
State - Next State	The rows and columns both indicate states, and the cell at the intersection of a row and column indicates the event that triggers a transition from the current (row) state to the next (column) state, the condition (or guard) of the event, and the effect of the transition.	Example State-Next State Table ^[830]

You can display a State Machine as a diagram or table, and as a table in one of three relationship formats.

How to:

To select the display format, follow the steps below:

Step	Action
1	Right-click on the diagram background to display the context menu
2	Select the Statechart Editor option
3	Select the appropriate display option: <ul style="list-style-type: none"> • Diagram • Table (State-Next State) • Table (State-Trigger) • Table (Trigger-State)

Learn More:

- [State Machine Table Options](#) ^[822]
- [State Machine Table Operations](#) ^[823]
- [State Machine diagram](#) ^[815]

6.3.4.1 State Machine Table Options

You can choose the State Machine table layout and set other options from the **State Machine Diagram: Options** dialog, which you display by either:

- Double-clicking on the State Machine table background or
- Right-clicking on the background and selecting the **State Table Options** context menu option

Field	Usage	See also
Table Format	Select the required table format: <ul style="list-style-type: none"> • State - Trigger: rows represent States, each state name in a left edge cell; columns represent Triggers, each trigger name in a column header cell; the intersection of a row and column identifies the Transition (if there is one); the Transition cell displays information about the next State and the condition (guard) of the Transition • Trigger - State: as above, except that rows represent triggers and columns represent states • State - Next State: both rows and columns represent states; intersection of row and column defines the transition (if there is one) from the row state to the column state. 	
Cell Size		
Transition Cell Width	Specify the width of the transition cells (that is, the column width).	
Transition Cell Height	Specify the height of the transition cells (that is, the row height).	
Left Edge Cell Width	Specify the width of the left edge (row title) cells.	
Top Edge Cell Height	Specify the height of the top edge (column title) cells.	
Cell Color		
State/Trigger Cell	Select the color of the row and column title cells.	
State/Trigger Enumeration	Select the color of the enumeration (row/column numbering) cells. You must select at least one of the Enable State Enumeration and Enable Event Enumeration checkboxes to set this color.	
Transition Cell	Select the color of the transition cells (in the main body of the table).	
Highlight Options		
Highlight Zones Related to Selected Transition	Highlight the cells for all elements involved in a selected transition - the initial state, the target state, and the trigger.	
Highlight Color	Select the color of the highlight.	
Use Different Color for Target State	Highlight the cell for the target element in a transition in a different color to the cell for the source element.	
Target Zone Color	Select the color of the highlight.	

Field	Usage	See also
Display Options		
Always Display an Empty State Zone	Add an empty row (and, on a <i>State - Next State</i> table, an empty column) to the end of the table. The title cell contains an ellipsis (...). You can click twice (not double-click) on the ellipsis to edit it and identify a new state. In this case, another empty state zone is automatically added.	
Enable State Enumeration	Add a cell to each state title cell, to number the state. Numbering starts at 0.	
Prefix	If required, type a prefix for the state number or delete the default S to have no prefix.	
Enable Event Enumeration	Add a cell to each event or trigger title cell, to number the event. Numbering starts at 0.	
Prefix	If required, type a prefix for the event number or delete the default E to have no prefix.	
Sample State Table	Display a preview of the table format as you define it.	
Advanced	Define diagram options. The State Machine Diagram Properties dialog displays.	Set Appearance Options
Restore Defaults	Reapply the State Table diagram default values.	
Apply	Apply changed options to the State Table diagram.	

6.3.4.2 State Machine Table Operations

As a State Machine table is a variant of a State Machine diagram, most of the operations for manipulating the data are the same as for State Machine diagrams. The operations specific to State Machine tables are described in the following topics:

Operation	See Also
Change State Machine Table Position	Change Position of State Machine Table
Change State Machine Table Size	Change Size of State Machine Table
Insert New State	Insert New State (and Substate)
Insert Trigger	Insert Trigger
Insert/Change Transition	Insert/Change Transition
Reposition State or Trigger Cells	Reposition State/Trigger Cells
Add Legend	Add Legend
Locate Cell in State Machine Diagram	Find Cell in State Machine Diagram
State Machine Table Conventions	State Machine Table Conventions
Export State Table To CSV File	Export State Table To CSV File

Learn More:

- [State Machine diagram](#) ^[815]
- [State Machine table](#) ^[821]

6.3.4.2.1 Change State Machine Table Position

If necessary, you can move the State Machine table around in the Diagram View.

How to:

To change the position of the State Machine table, follow the steps below:

Step	Action
1	Press (Ctrl+A) or double click on the top left cell to select the whole State Machine table
2	Drag and drop the State Machine table to the required position Alternatively, use (Shift) + (←) , (↑) , (→) or (↓) to move the State Machine table

6.3.4.2.2 Change State Machine Table Size

There are three ways to change the size of the State Machine table:

Action	See Also
Change the cell size on the State Machine Diagram: Options dialog	State Machine Table: Options ^[822]
Press (Ctrl+A) or double click on the top left cell to select the whole State Machine table, then press (Ctrl) + (←) , (→) , (↑) or (↓) to change the size	
Select the State Machine table, then drag the shape handles to change the size	

6.3.4.2.3 Insert New State

Action	Description
Insert a new State in the State Machine table	<p>You can insert a new State in the State Machine table, using one of following methods:</p> <ul style="list-style-type: none"> • In the top left cell in the State Machine table, move the cursor to the word State to display a + at the end of the word; click on the + to create a new State • Right-click in the top left cell in the State Machine table to display the context menu, and select the Add State menu option • Right-click on an existing State cell in the State Machine table to display the context menu, and select the <ul style="list-style-type: none"> • Insert New State Before option to insert a new State before the current State, or • Insert New State After option to insert a new State after the current State • Click on an existing State cell in the State Machine table, and press (Insert) to create and insert a new State above the selected State • In the Toolbox, on the State Elements page, click on an element and then click

	<p>on:</p> <ul style="list-style-type: none"> the diagram background to add a new State to the end of the table, or an existing State cell to add the new State just above it. <p>From the State Elements page of the Toolbox you can insert State, Initial, Final, Entry, Exit and Terminate elements.</p>
Add a Substate to a selected State	<p>To add a Substate to a selected State, follow the steps below</p> <ol style="list-style-type: none"> Right-click on the required State cell in the State Machine table; the context menu displays Select the Add Substate menu option; Enterprise Architect adds the Substate to the State <p>Note: If the selected State does not allow a Substate, then the Add Substate menu option is grayed out.</p> <p>You can also drag one existing State over another. If the second State allows Substates, the dragged State then becomes its Substate.</p> <p>Similarly, you can change the parent State of a Substate by dragging the Substate from the original parent State to a different State.</p>
Remove the parent relation of a Substate and make it a separate State	<p>To remove the parent relation of a Substate and make it a separate State, follow the steps below</p> <ol style="list-style-type: none"> Right-click on the Substate in the State Machine table. The context menu displays. Select the Remove Parent Relation menu option. The Substate cell becomes a State cell. <p>You can also drag and drop the Substate onto the top left cell of the State Machine table. The dragged Substate again becomes a State cell.</p>

6.3.4.2.4 Insert Trigger

If the State Machine table format is either *State-Trigger* or *Trigger-State*, you can use one of the following methods to insert a new Trigger:

Method	Action
a	In the top left cell in the State Machine table, move the cursor to the word Event to display a + at the end of the word; click on the + to create a new Trigger
b	In the top left cell in the State Machine table, right-click to display the context menu and select the Add Trigger menu option to create a new Trigger
c	Select an existing Trigger in the State Machine table, then press (Insert) to insert a new Trigger before the existing Trigger
d	Click on an existing Trigger in the State Machine table, right-click to display the context menu and select either the: <ul style="list-style-type: none"> Insert New Trigger Before option to insert a new Trigger before the current Trigger, or Insert New Trigger After option to insert a new Trigger after the current Trigger

6.3.4.2.5 Insert/Change Transition

Action	Description	See Also
Insert a new	You can insert a new Transition using one of the following	

Transition	<p>methods:</p> <ul style="list-style-type: none"> • Right-click on the cell in which to create a Transition, to display the context menu <ul style="list-style-type: none"> • If the State Machine table format is State-Trigger or Trigger-State, the context menu lists the States you can choose as the target of the Transition; click on the required State name to create the Transition • If the State Machine table format is State-Next State, click on the Insert Transition menu option to create the Transition. • In the State Relationships page of the Toolbox, select the Transition element, then click on the cell in the State Machine table in which to create the Transition. Double-click on the Transition to define it in the Transition Properties dialog. 	
Change the Transition	As for the State Chart diagram, to change the properties of a Transition double-click the Transition cell and edit the details on the Transition Properties dialog.	State Machine Diagrams ^[815]
Change Transition States	<p>You can change the source and target of the Transition by right-clicking the Transition and selecting the Advanced Set Source and Target context menu option.</p> <p>Alternatively, you can change the Transition source, target or Trigger by clicking on the Transition and dragging it to a different cell.</p> <p>If the State Machine table format is either State-Trigger or Trigger-State, you can change the target state of a transition by:</p> <ol style="list-style-type: none"> 1. Highlighting the target state name in the Transition cell and clicking on it to display a list of the states in the table. 2. Clicking on the preferred target state name. 	
Highlight States and Trigger Related to Transition	<p>You can select options to highlight the source State, target State and Trigger cells associated with a Transition, using the Highlight Options panel on the State Machine Diagram: Options dialog. When you click on the Transition cell its associated State and Trigger cells are highlighted.</p> <p>Alternatively, click on the Transition cell and press and hold (L).</p>	State Machine Diagram: Options ^[822]

6.3.4.2.6 Reposition State or Trigger Cells

You can change the position of a selected State or Trigger cell in one of the following ways:

- Right-click on the State or Trigger title cell and select the appropriate **Order | Move xxx** context menu option
- Click on the cell and press (**Shift**) + (**→**), (**←**), (**↑**) or (**↓**).

6.3.4.2.7 Add Legend

You can add a simple legend to any State Machine Table cell that has no transition. The two legend symbols are:

- **I** - Ignore
- **N** - Never Happen

How to:

To assign a legend symbol to a State Machine Table cell, follow the steps below:

Step	Action
1	<p>Click on the cell to which to assign the legend and press:</p> <ul style="list-style-type: none"> • (I) to insert the Ignore legend, or • (N) to insert the Never Happen legend <p>The required symbol displays in the center of the cell</p>

Alternatively:

Step	Action
1	<p>Right-click on the cell to which to assign the legend</p> <p>The context menu displays</p>
2	<p>Select the appropriate menu option:</p> <ul style="list-style-type: none"> • Legend Ignore • Legend Never Happen <p>The required symbol displays in the center of the cell</p>

Notes:

- To remove a legend symbol from a cell, either:
 - Click on the cell and press **(Delete)**, or
 - Right-click on the cell and select the **Legend | Remove Legend** context menu option

6.3.4.2.8 Find Cell in State Machine Diagram

Locate	Description
in State Chart	<p>On the State Machine table you can select a State or Trigger element and locate it in a State Machine diagram, by selecting the Find Locate in State Chart context menu option. Enterprise Architect switches to the State Machine diagram and highlights the selected element. You can locate a Transition relationship in a similar way, by selecting the Locate in State Chart context menu option.</p> <p>A Trigger on a State Machine table might or might not exist on the corresponding State Machine diagram. If the Trigger does not exist on the State Machine diagram, the Locate in State Chart option is disabled.</p>
in State Table	<p>Conversely, on the State Machine diagram, you can select a State or Trigger element and locate it on the corresponding State Machine table, by selecting the Find Locate in State Table context menu option. Enterprise Architect switches to the State Machine table and highlights the selected element. You can locate a Transition relationship in a similar way, by selecting the Locate in State Table context menu option.</p>

6.3.4.2.9 State Machine Table Conventions

Trigger

- Deleting a Trigger removes it completely from the model, therefore you cannot UNDO a Trigger deletion
- There is a <None> column at the end of the **Event** heading row. This is for Transitions that have no Trigger information.

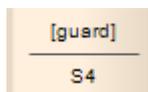
State

From the **Toolbox** you can insert the following *State* element types only (although the State Machine table might pick up and display other types, such as *Submachine State*):

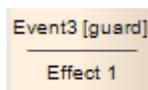
- State
- Initial
- Final
- Entry
- Exit
- Terminate.

Transition

The Transition cell displays its properties in one of two ways, depending on the State Machine table format. If the State Machine table format is *State - Trigger* or *Trigger - State*, the Transition cell displays the *Guard* and *Target* as shown below:



If the State Machine table format is *State - Next State*, then the Transition cell displays the *Trigger*, *Guard* and *Effect* as shown below:



The State Machine table enables you to edit the *Guard* and *Effect* in place. If the *Guard* or *Effect* is empty for your selected Transition cell, the cell displays an ellipsis (...) instead. Click twice (not double-click) on the ellipsis to type in the *Guard* and *Effect* names.

6.3.4.2.10 Export State Table To CSV File

To export a State Machine Table to a CSV file, follow the steps below:

Step	Action
1	Open the required State Machine Table
2	Right-click on the diagram background and select the Export Statechart to CSV file context menu option The Save As browser dialog displays
3	Select the appropriate directory location and type in the .CSV filename

4	Click on the Save button
---	---------------------------------

6.3.4.3 Example State-Trigger Table

State - Trigger:

The rows indicate the current states and the columns indicate trigger events (or the other way around if you prefer, in a **Trigger - State** format). The cell at the intersection of a row and column identifies the target state in the transition if the trigger occurs, and the condition (or guard) of the transition.

		Trigger		Event1	Event2	Event3	Event4	<None>
		State		E0	E1	E2	E3	E4
Initial	S0							S1
State1	S1						S2	
State2	S2	S6	$\frac{[\text{Guard}]}{S4}$					
	SubState1	S3	S4					
	SubState2	S4			$\frac{[\text{Cond}]}{S2}$			
	SubState3	S5						
State3	S6							S7
Final	S7							

6.3.4.4 Example State-Next State Table

State - Next State: The rows and columns both indicate states, and the cell at the intersection of a row and column indicates the event that triggers a transition from the current (row) state to the next (column) state, the condition (or guard) of the event, and the effect of the transition.

Next State		Initial		State1			State2			State3	Final
							SubState1	SubState2	SubState3		
		State		S0	S1	S2	S3	S4	S5	S6	S7
Initial		S0		_____							
State1		S1			Event4 _____						
State2		S2					Event2 [Guard] _____		Event1 _____		
	SubState1	S3					Event2 _____				
	SubState2	S4			Event3 [Cond] _____						
	SubState3	S5									
State3		S6									_____
Final		S7									

6.3.5 Timing Diagram

A Timing diagram defines the behavior of different objects within a time-scale. It provides a visual representation of objects changing state and interacting over time.

Use To:

- to define hardware-driven or embedded software components; for example, those used in a fuel injection system or a microwave controller.
- to specify time-driven business processes.

Example Diagram: [Example Timing Diagram](#) ^[83]

Tools: Select Timing diagram elements and connectors from the Timing pages of the **Toolbox**. Click on the elements and connectors below for more information.

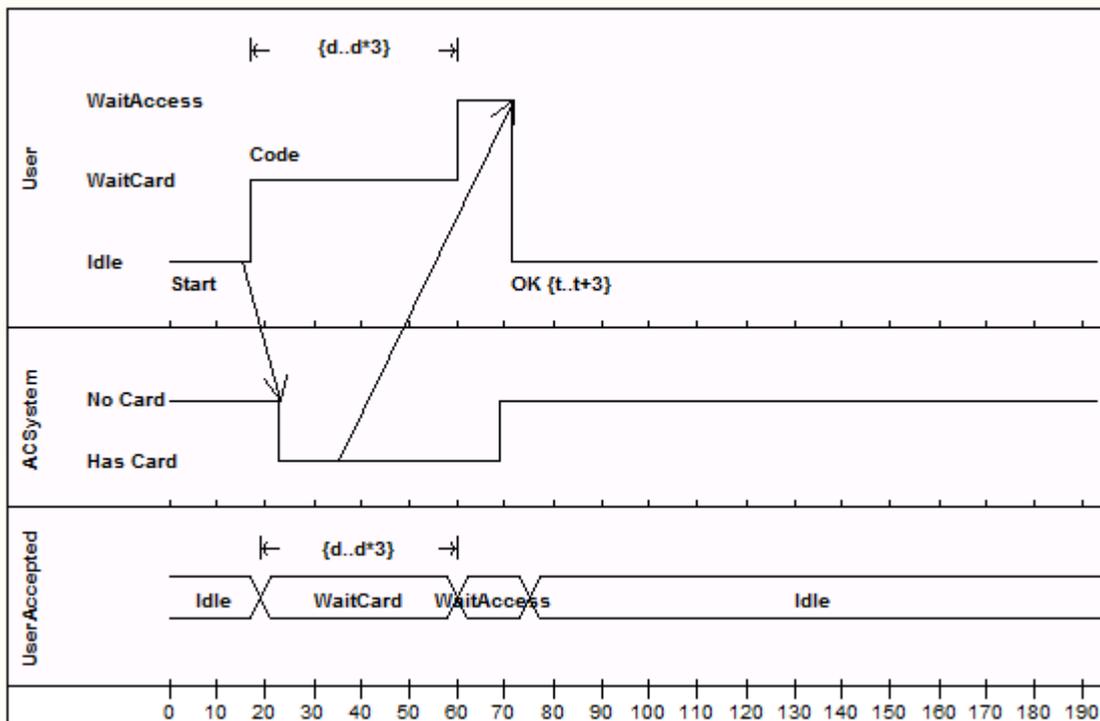
Timing Diagram Elements	Timing Diagram Message
 State Lifeline	 Message
 Value Lifeline	
 Message Label	
 Message Endpoint	
 Diagram Gate	

Learn More:

- [Create a Timing Diagram](#)^[832]
- [Set a Time Range](#)^[832]
- [State Lifeline](#)^[923]
- [Value Lifeline](#)^[938]
- [Edit a Timing Diagram](#)^[833]
- [Time Intervals](#)^[843]
- [Message \(Timing Diagram\)](#)^[999]
- [Timing](#)^[557] [pages](#)^[557]

6.3.5.1 Example Timing Diagram

An example of a Timing diagram is shown below:



6.3.5.2 Create a Timing Diagram

To create a Timing diagram, follow the steps below:

Step	Action
1	Right-click on a package in the Project Browser The context menu displays
2	Select the Add Add Diagram menu option The New Diagram dialog displays
3	In the Select From panel, select UML Behavioral
4	In the Diagram Types panel, select Timing
5	Click on the OK button The Diagram View displays, on which you create the Timing elements for the diagram

Learn More:

- [Set a Time Range](#)^[832]
- [Edit a Timing Diagram](#)^[833]

6.3.5.3 Set a Time Range

To set a time range before adding Lifeline elements to your Timing diagram, follow the steps below:

Step	Action
1	Right-click on the diagram The context menu displays
2	Select the Set Timeline Range option The Set Timeline Range dialog displays
3	In the Start Time and End Time fields, type the numeric values for the start and end points of the timeline; for example, set the range 0 to 100 The start time must be less than the end time
4	In the Time Units field, type the unit in which the time is measured; for example, seconds or minutes
5	If it is not necessary to show the time range on the diagram, select the Suppress In Diagram checkbox
6	Click on the OK button If you have not suppressed it, the time range displays underneath the Lifeline elements that you create on the diagram

6.3.5.4 Edit a Timing Diagram

On a Timing Diagram, you can add State Lifeline elements and Value Lifeline elements. You can maintain the states and transitions on these Lifeline elements either on the diagram itself or via the Configure Timeline dialog.

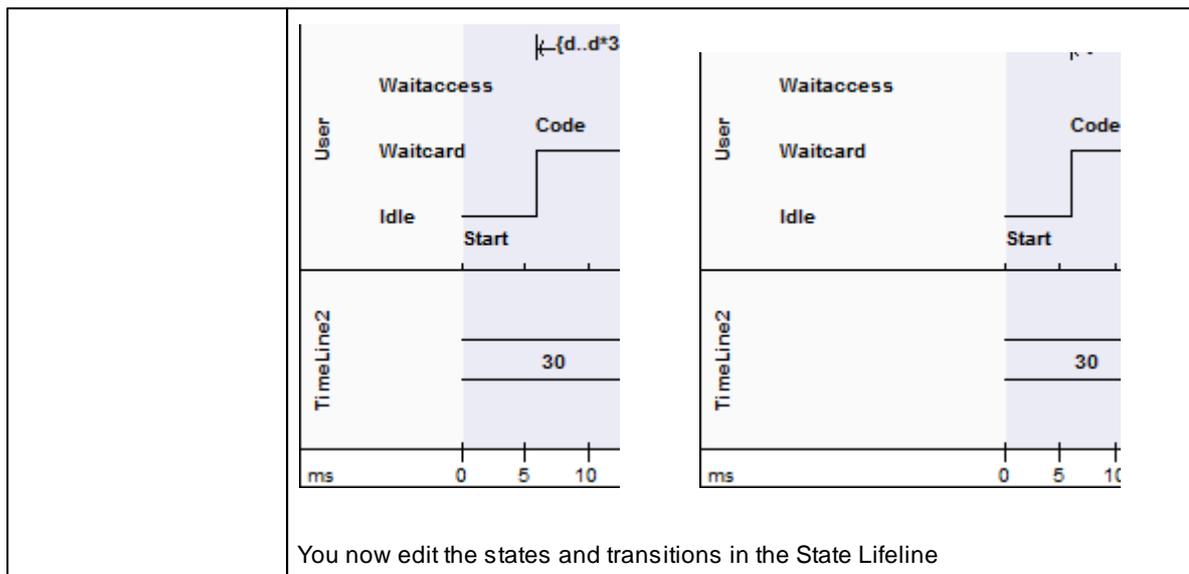
Learn More:

- [Add and Edit a State Lifeline Element](#)^[833]
- [Add States to a State Lifeline](#)^[834]
- [Edit States in a State Lifeline](#)^[835]
- [Delete States in a State Lifeline](#)^[835]
- [Edit Transitions in a State Lifeline Element](#)^[835]
- [Add and Edit a Value Lifeline Element](#)^[836]
- [Add States in a Value Lifeline Element](#)^[836]
- [Edit Transitions in a Value Lifeline Element](#)^[836]
- [Configure Timeline dialog - States Tab](#)^[840]
- [Configure Timeline dialog - Transitions Tab](#)^[842]

6.3.5.4.1 Add and Edit State Lifeline

From the Timing elements page of the Toolbox drag a State Lifeline element onto your diagram. The element displays on the diagram.

Task	Action
Define the name of the State Lifeline	<ol style="list-style-type: none"> 1. Right-click on the element; the context menu displays 2. Select the Other Properties option; the Timeline <name> dialog displays, showing the General tab 3. Overtyping the Name field 4. Click on the Apply button and the OK button
Sizing and Scale	<p>In the top left corner of a selected Lifeline element are the left and right quick sizing buttons ( )</p> <p>These buttons increase or decrease the width of the Lifeline element, which in turn controls the scale width of each time unit; by increasing the width of the element you increase the resolution when adding transitions, which makes them easier to edit</p> <p>In order to edit the State Lifeline element, you must click on it to select it</p>
Set Timeline Start Position	<p>You might require more space at the start of your timelines; for example, to use long state names</p> <p>To insert more space in all the timelines on a diagram, follow the steps below:</p> <ol style="list-style-type: none"> 1. Right-click on the diagram background to display the context menu 2. Select the Set Timeline Start Position menu option; the Set Timeline Start Position dialog displays 3. The Value 80 to 300 field defaults to 80 as the minimum distance in pixels between the start of the timeline element and the start of the timeline itself; type a new value up to 300 pixels and click on the OK button to increase the space at the start of the timeline, as shown in the following diagrams (80 pixels and 150 pixels respectively)



Learn More:

- [Add States to a State Lifeline](#) ^[834]
- [Edit States in a State Lifeline](#) ^[835]
- [Delete States in a State Lifeline](#) ^[835]
- [Edit Transitions In State Lifeline](#) ^[835]

6.3.5.4.2 Add States to a State Lifeline

To add States to a State Lifeline, follow the steps below:

Step	Description	See Also
1	Click on the <i>State Lifeline</i> element The New State button () and Edit States button () display at the bottom left of the element	
2	Click on the New State button The New State dialog displays	
3	In the State field, type the name of the state	
4	Click on the OK button You must add at least two states; for example, On and Off	
5	As you add states, increase the height of the element by dragging a handle-box () on the edge of the element You can also add states using the States tab of the Configure Timeline dialog; add either: <ul style="list-style-type: none"> • Discrete states to the Timeline, or • A continuous range of numeric states 	Add a New State ^[840] Numeric Range Generator ^[841]

6.3.5.4.3 Edit States in a State Lifeline

To edit States in a State Lifeline, follow the steps below:

Step	Description	See Also
1	Click on the State Lifeline element and click on the required state The Edit State dialog displays	
2	In the State field, change the name as required	
3	Click on the OK button	
4	If necessary, change the order of the states by either: <ul style="list-style-type: none"> Clicking on the up or down arrows (⬆ and ⬇) beside each state name, or Right-clicking on the state name and selecting the Move Up or Move Down context menu options You can also edit the states using the States tab of the Configure Timeline dialog	States ^[840]

6.3.5.4.4 Delete States in a State Lifeline

To delete States in a State Lifeline, follow the steps below:

Step	Description
1	Right-click on the state name The context menu displays
2	Select the Delete option

Alternatively:

Step	Description
1	Click on the State Lifeline element
2	Hold down (Ctrl) and move the cursor over the state name The cursor changes form (↔)
3	Click the mouse button The state name is deleted

6.3.5.4.5 Edit Transitions In State Lifeline

Task	Action	See Also
Add and Move Transitions	After you have added states, you can add transitions via the diagram	Add and Move Transitions ^[836]
Change the Transition	Move the cursor over one or other of the vertical	

Time	<p>transition lines and drag the line left or right to change the time of the transition</p> <p>While on the line, the cursor shape changes to the horizontal movement cursor (←→)</p>	
Merge Transitions	<p>If necessary, you can 'push' a transition to merge it with the next or previous transition point on any Lifeline element on the diagram</p> <p>Position the cursor off the appropriate side of the transition line; the cursor changes form (← or →)</p> <p>Click the mouse button; the system locates the nearest transition in the required direction, on any element on the diagram, and merges the current transition with that transition</p>	
Delete Transitions	<p>Transitions are automatically deleted when you move the transition to the same state as the previous transition state, and release the cursor</p> <p>Alternatively, right-click on the transition line to display the context menu, and select the Delete menu option</p>	

6.3.5.4.5.1 Add and Move Transitions

After you have added states, you can add transitions via the diagram. As you move the cursor over the timeline, the cursor changes to one of three shapes.

Task	Action	See also
The move cursor (↕)	<p>Displays when it is directly over the timeline</p> <p>Hold down the mouse button and drag the line to move the timeline to a state above or below the current position; you can move the transition more than one state up or down, if necessary</p>	
The new transition up cursor (↑)	<p>Displays when it is just below the timeline, and there is another state above the line</p> <p>Press and hold (Alt); the cursor changes (↑)</p> <p>Click to create a new transition to the state above the line</p> <p>To push the transition up more than one state, move the cursor onto the line and drag it up</p> <p>The transition is for one interval unit; you can make it longer by changing the transition time</p> <p>If you do not hold (Alt), the cursor does not change and the whole timeline from the transition onwards moves up</p>	Edit Transitions In State Lifeline [835]
The new transition down cursor (↓)	<p>Displays when it is just above the transition line, and there is another state below the line</p> <p>Press and hold (Alt); the cursor changes (↓)</p>	Edit Transitions In State Lifeline [835]

	<p>Click to create a new transition to the state below the line</p> <p>To push the transition down more than one state, move the cursor onto the line and drag it down</p> <p>The transition is for one interval unit; you can make it longer by changing the transition time</p> <p>If you do not hold (Alt), the cursor does not change and the whole timeline from the transition onwards moves down</p>	
--	--	--

As you move the cursor over the vertical line of a transition, the time at which the transition occurs displays next to the line.

How to:

To edit Transitions, follow the steps below:

Step	Action
1	Click directly on the appropriate transition line, after the transition begins Alternatively, right-click on the transition line to display the context menu, and select the Edit menu option The Edit Transition dialog displays; the fields in this dialog are all optional
2	In the At Time field, type the point on the timescale at which the transition occurs
3	In the Transition To field, type the name of the state to which the transition occurs
4	In the Event field, type the name of the event that the transition represents; this displays on the Timeline element just above the transition line
5	In the Duration Constraint field, type any constraint on the duration of the transition; this displays on the Timeline element, along the top of the element over the transition
6	In the Time Constraint field, type any constraint on the start of the transition This displays on the Timeline element at the start of the transition
7	Click on the OK button

Notes:

- Once Event, Duration Constraint or Time Constraint are displayed on the diagram, you can edit them directly by clicking on them to display their specific dialog
- You can delete them by pressing and holding (**Ctrl**) as you click on them; the cursor changes form when you press (**Ctrl**)
- You can also edit transitions using the Transitions tab of the Configure Timeline dialog

Learn More:

- [Transitions](#)^[842]

6.3.5.4.6 Add and Edit Value Lifeline

From the Toolbox drag a Value Lifeline element onto your diagram. The element displays on the diagram.

How to:

To edit the Value Lifeline name, follow the steps below:

Step	Action
1	Right-click on the element The context menu displays
2	Select the Other Properties option The Timeline <name> dialog displays, showing the General tab
3	Overtyping the Name field
4	Click on the Apply button and the OK button

Sizing and Scale

In the top left corner of a selected Lifeline element are the left and right *quick sizing* buttons ( ). These buttons increase or decrease the width of the Lifeline element, which in turn controls the scale width of each time unit. By increasing the width of the element you increase the resolution when adding transitions, which makes them easier to edit.

Learn More:

- [Value Lifeline](#) ^[938]
- [Add States in a Value Lifeline](#) ^[838]
- [Edit Transitions in a Value Lifeline](#) ^[838]

6.3.5.4.7 Add States In Value Lifeline

Adding States to a Value Lifeline is similar to adding states to a State Lifeline element.

For a Value Lifeline, only the first state displays on the diagram. The other states are added to a list to access when creating transitions; they only display on the Lifeline element as you create transitions to those states.

You can only edit or delete states in a Value Lifeline element using the **States** tab of the **Configure Timeline** dialog.

Learn More:

- [Add States to a State Lifeline](#) ^[834]

6.3.5.4.8 Edit Transitions In Value Lifeline

How to:

To add transitions to the states on a Value Lifeline element, via the diagram, follow the steps below:

Step	Action
1	Move the cursor above the transition line The cursor changes form ()
2	Click the mouse button The New Transition Event dialog displays
3	In the Transition To field, click on the drop-down arrow and select a state from the list of available states; this displays on the Lifeline element within the transition box The remaining fields on the dialog are optional
4	In the Event field, type the name of the event that the transition represents; this displays on the Lifeline element just below and at the start of the transition line
5	In the Duration Constraint field, type any constraint on the duration of the transition; this displays on the Lifeline element, along the top of the element over the transition
6	In the Time Constraint field, type any constraint on the start of the transition This displays on the Lifeline element at the start of the transition, just after the Event name
7	Click on the OK button to create the new transition

To edit a transition, follow the steps below:

Step	Action
1	Click on the state name in the transition Alternatively, right-click on the state name to display the context menu, and select the Edit menu option The Edit Transition dialog displays, which is the same as the New Transition Event dialog, except that the At Time field is enabled
2	If necessary, overwrite the At Time field to define a different start point You cannot change the At Time field for the first state in the timeline; this is always 0
3	Edit the remaining fields as necessary
4	Click on the OK button to save the changes

To change the Transition Time, follow the steps below:

Step	Action
1	To change the start or end time of a transition, click on the start or end point of the transition and drag it to the new position While on the line, the cursor shape changes to the horizontal movement cursor ()

To delete Transitions, follow the steps below:

Step	Action
1	To delete a transition, press and hold (Ctrl) and click on the transition state name While you hold (Ctrl) on the transition state name, the cursor changes form (↖). Alternatively, right-click on the state name to display the context menu, and select the Delete menu option

6.3.5.4.9 Configure Timeline - States

You can manage states using the States tab of the Configure Timeline dialog. To display this, either:

- Double-click on the Lifeline element
- Right click on the Lifeline element and, from the context menu, select the **Properties** option, or
- On a Value Lifeline, click on the **Edit States** button ()

The Configure Timeline dialog defaults to the States tab.

All states currently defined for the Lifeline element are listed in the **States** panel.

How to:

To add a new State, follow the steps below:

Step	Action
1	In the State Name field, type the name of the first new state in the Lifeline element; for example, WaitState
2	Click on the Save button The state is added to the States panel and (for a State Lifeline Element) to the diagram
3	Click on the New button
4	In the State Name field, type the name of the next state in the Lifeline element
5	Repeat steps 2 to 5 until you have added all required states (you must add at least three to the Lifeline element)
6	When you have added all the required states, click on the OK button to close the Configure Timeline dialog

To edit an existing state, follow the steps below:

Step	Action
1	Click on the state in the States: list
2	In the State Name field, change the name of the state
3	Click on the Save button

To delete an existing State, follow the steps below:

Step	Action
1	Click on the state in the States: list
2	Click on the Delete button

To change the order of States, follow the steps below:

Step	Action
1	Click on the state in the States: list.
2	Click on the  or  buttons to move the state up or down the sequence

See Also:

- [Numeric Range Generator](#)^[841]

6.3.5.4.10 Numeric Range Generator

You can also use the Configure Timeline dialog to create a range of states having numeric values to be applied to the Timeline.

Important: This operation deletes all existing states and transitions for the Timeline element.

How To:

To create a range of states having numeric values, follow the steps below:

Step	Action
1	Double-click on the Lifeline element The Configure Timeline dialog displays
2	Click on the Create Continuous Numeric States button The Numeric Range Generator dialog displays
3	In the High Value and Low Value fields, type the upper and lower values of the range
4	In the Step Value field, type the increase interval Nonsense values do not parse; Low Value must be less than High Value , and Step Value must be a positive value smaller than the total range
5	In the Units field, type the name of the measurement unit; for example, minutes
6	Click on the OK button Enterprise Architect displays a warning that existing states and transitions are to be deleted
7	Click on the Yes button

	The Configure Timeline dialog redisplay, with the defined range of states listed in the States panel
8	<p>Click on the OK button</p> <p>For a:</p> <ul style="list-style-type: none"> • Value Lifeline, the first state is shown on the Timeline for the full time range of the Timeline • State Lifeline, the range of states is displayed as the y-axis of the Timeline

6.3.5.4.11 Configure Timeline - Transitions

You can also manage transitions using the Transitions tab of the Configure Timeline dialog. To display this, either:

- Double-click on the Lifeline element
- Right click on the Lifeline element and, from the context menu, select the **Properties** option, or
- On a Value Lifeline, click on the **Edit States** button ().

The Configure Timeline dialog defaults to the States tab. Click on the Transitions tab.

All transitions defined for the Timeline element are listed in the Transition Points panel.

How To:

To add a new transition, follow the steps below:

Step	Action
1	Click on the New button
2	In the New Transition panel, type the details of the transition
3	Click on the Save button

To edit a transition, follow the steps below

Step	Action
1	Click on a transition in the list
2	In the Edit Transition panel, edit the fields for the transition as required
3	Click on the Save button

To delete a transition, follow the steps below

Step	Action
1	Click on a transition in the list
2	Click on the Delete button

	The transition is removed from the dialog and the Lifeline
3	Click on the OK button

6.3.5.5 Time Intervals

You create and manage Time Intervals using the *Interval Bar* (the pale line along the top of each selected Lifeline element). Time Intervals enable you to perform various operations on transitions, such as copy and paste. They also enable you to compress sections of the timeline so that they are not visible.

Each Time Interval displays across all Timeline elements down to the last element on the diagram.

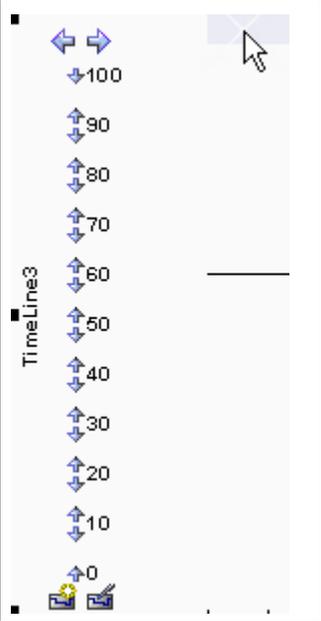
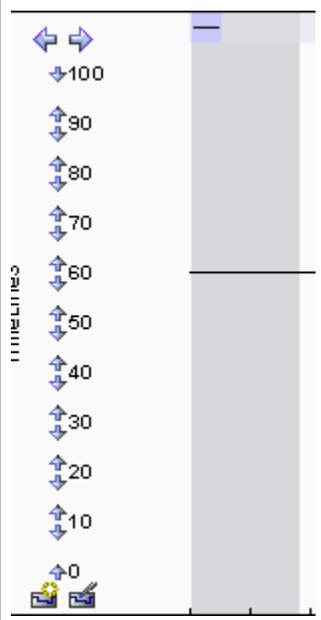
Action	Description	See
Create Time Intervals	You can create a Time Interval using the Interval Bar - Context Menu, using the Interval Bar - (Shift) key, or using the Timeline - Context menu.	Create Time Intervals ^[843]
Compress Time Intervals	You can compress Time Intervals to conserve space on long timelines.	Compress Time Intervals ^[845]
Select Time Intervals	There are a number of ways to select Time Intervals for performing other operations.	Select Time Intervals ^[846]
Move Time Intervals	To move a Time Interval, move the cursor over the Interval bar within the Time Interval, hold down the mouse button and drag the interval left or right Time Intervals can meet, but cannot overlap.	
Resize Time Intervals	To resize a Time Interval, move the cursor over the Interval Bar at the start or end edge of the Time Interval, hold down the mouse button and move the edge left or right Time Intervals can meet, but cannot overlap.	
Delete Time Intervals	To Delete Time Intervals, select each Time Interval to be deleted and press (Delete) Deleting the Time Interval does not delete transitions within that interval	

6.3.5.5.1 Create Time Intervals

How to:

To create a Time Interval using the Interval Bar context menu, follow the steps below:

Step	Description	

1	<p>Right-click on the Interval Bar at approximately the point at which to start or finish the Time Interval</p> <p>The context menu displays</p>	
2	<p>Select the Create Time Interval option</p> <p>The Time Interval displays down all the timeline elements, as a narrow pale band with a blue compression box at the top</p>	
3	<p>Move the cursor to the edge of the Time Interval in the Interval Bar so that the cursor changes to the drag form (↔) and drag the edge to the correct start or end point</p>	

To create a Time Interval using the Interval Bar and (**Shift**) key, follow the steps below:

Step	Description	See also
1	<p>Move the cursor over the Interval Bar and press (Shift)</p> <p>The cursor changes shape (↔)</p>	
2	<p>Click to create the Time Interval</p>	
3	<p>Move the cursor to the edge of the Time Interval in the Interval</p>	

	Bar so that the cursor changes to the drag form (↔) and drag the edge to the correct start or end point	
--	---	--

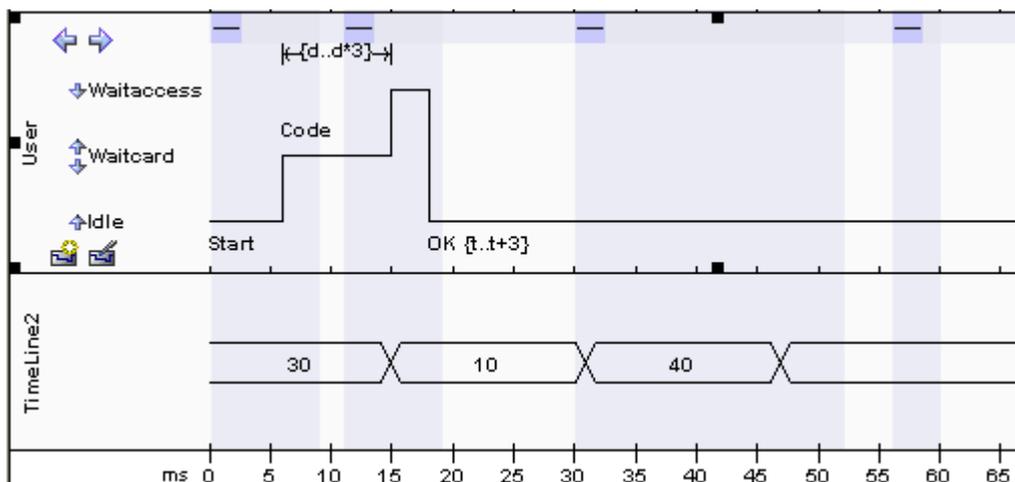
To create a Time Interval using the Timeline context menu, follow the steps below:

Step	Description	See Also
1	Right-click on the timeline just after a transition The context menu displays	
2	Click on the Select menu option Enterprise Architect creates a Time Interval covering the period from the selected transition up to the next transition If there are other Time Intervals in this period, Enterprise Architect replaces them with the single Time Interval for the transition state; you should consider this when creating the Time Interval, as it extends across the other Timeline elements in the diagram A value of this method is that it creates a Time Interval for a period in which no transitions occur, which could be lengthy; you can then compress this Time Interval to hide the period of inactivity	Compress Timeline ^[845]

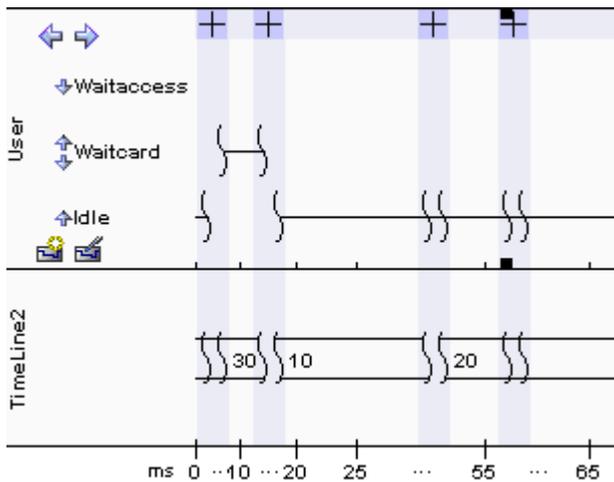
6.3.5.5.2 Compress Time Intervals

You can compress Time Intervals to conserve space on long timelines.

Uncompressed Time Intervals



Compressed Time Intervals



Notice:

Item	Description
	The compression toggle boxes: <ul style="list-style-type: none"> • is expanded, click on this to compress the selected time interval • is compressed, click on this to expand the selected time interval again.
	The compressed sections of the timelines themselves, in all elements. If there is space between the paired symbols, there are transitions within the compressed section. If the timeline continues through the paired symbols there are no transitions in the compressed section.
25 ... 55	The compressed sections in the time range underneath the elements.

You can also compress and expand Time Intervals using context menu options; see [Time Interval Operations on Transitions](#) ^[847].

6.3.5.5.3 Select Time Intervals

Task	Action	See Also
Select a Time Interval across all elements on the diagram	Click on the Interval Bar within the Time Interval	
Select a number of individual Time Intervals	Press and hold (Ctrl) while clicking on the Interval Bar within each Time Interval	
Select all Time Intervals in a range	Click on the Interval Bar within the first Time Interval in the range, then press and hold (Shift) and click on the Interval Bar within the last Time Interval in the range All Time Intervals between the two are selected	

After you have selected one or more Time Intervals, you can modify the selection in the following ways:

Task	Action	See Also
Exclude Lifeline elements from the selection	Press and hold (Ctrl) and click on any part of the selection within that element Repeat the step to toggle the selection and re-include the element	Toggle Interval Selection ^[847]
Select only one Lifeline element and exclude all others	Press and hold (Shift) and click on any part of the selection within that element	

6.3.5.5.4 Time Interval Operations

This topic discusses how you select and update specific Time Intervals.

Right-click on the Interval Bar within an interval. A context menu displays providing the following options:

Option	Action	See also
Select Interval Deselect Interval	Select the Time Interval or, if the interval is already selected, deselect it You can select several Time Intervals in this way, accessing the menu separately on each interval	Select Time Intervals ^[848]
Toggle Interval Selection	Switch the selection or deselection of the Time Interval within the selected Timeline element You select or deselect a Time Interval across all Timeline elements, but the Toggle option acts only on the element in which you access the menu	Select Time Intervals ^[848]
Compress Interval	Compress the Time Interval, and hide all transitions within that Time Interval This is also useful for hiding long sections of inactivity on the time line	Compress Timeline ^[848]
Remove Interval	Delete the Time Interval	
Copy	Copy the transitions for all selected Time Intervals	
Cut	Copy and delete the selected transitions from the diagram	
Cut and Remove Time	Copy and delete the transitions that lie in the selected Time Intervals from the diagram This option also removes time from the timeline, the amount being the duration of the Time Interval All transitions and Time Intervals to the right of the selected time interval are moved left	
Delete	Delete the selected transitions from the diagram	
Delete and Remove Time	Delete the transitions that lie in the selected Time Intervals from the diagram This option also removes time from the timeline, the amount being the duration of the Time Interval All transitions and Time Intervals to the right of the current Time Interval are moved left	

Option	Action	See also
Insert Time	Add time to the timeline and move all transitions and time intervals to the right Also expand the duration of the current Time Interval	

Compress Timeline:

The Compression toggle boxes and **Compress Interval** menu option operate on the Time Interval and compress the timeline and all transitions within the Interval. You have an alternative option that operates on the timeline and compresses a single transition state.

1. Right-click on the timeline (rather than the Interval Bar) just after a transition. The context menu displays.
2. Click on the **Compress** menu option. Enterprise Architect creates a new Time Interval covering the period from the selected transition up to the next transition, and then compresses that Time Interval.

If there are other Time Intervals in this period, Enterprise Architect replaces them with the single Time Interval for the transition state. You should consider this when creating and compressing the Time Interval, as it extends across the other Timeline elements in the diagram.

A value of this method is that it creates a Time Interval for a period in which no transitions occur, which could be lengthy, and then compresses this Time Interval to hide the period of inactivity.

All Time Intervals in the Diagram:

To create a new Time Interval or work across all Time Intervals in the diagram, right-click on the Interval Bar between Time Intervals. A context menu displays, providing the following options (The **Paste ...** menu options become active after transitions have been copied).

Menu Option	Action	See also
Create Time Interval	Create a single Time Interval	Create Time Intervals [843]
Expand all Time Intervals	Expand all Time Intervals over the whole diagram	
Compress all Time Intervals	Compress all Time Intervals over the whole diagram	
Paste Combine	Paste copied transitions over any existing transitions within the copied time frame The diagram does not allow two consecutive transitions to the same state, and removes the second transition automatically	
Paste Remove	Delete all the transitions and then pastes the copied transition within the copied time frame	
Paste Insert	Insert time, moving all transitions and Time Intervals to the right to make room to paste in the copied transitions	
Insert Time	Add time to the timeline and move all transitions and Time Intervals to the right This option does not change the duration of any Time Interval	

To copy and paste transitions from one timeline element to another, follow the steps below:

Step	Action	See Also
1	Press and hold (Shift) and select the Timeline element within a Time Interval to copy or cut	
2	Right-click on the Interval Bar (it doesn't matter which element you select) The context menu displays	
3	Copy or cut the transitions You can also cut and remove time	Copy or cut ^[847]
4	Select the timeline to paste transitions to and right-click on the Interval Bar The context menu displays	
5	Select one of the paste operations Note that states are created if they don't already exist in the timeline Any states that don't exist in the Timeline element you are pasting to are created Any new states created might be in the wrong order; you can change the order via the diagram quick buttons	Quick Buttons ^[833]

To shift transitions within a selected Time Interval or multiple selected Time Intervals, follow the steps below:

Step	Action	See Also
1	Select all the Time Intervals containing the transitions to be shifted	Select Time Intervals ^[846]
2	Press and hold (Shift) and click on the Interval Bar (it doesn't matter which Timeline element you select), and move the transition left or right You cannot drag transitions over other transitions; the move stops when the moved transition collides with a stationary transition If you have collision problems, use (Shift) + select to shift transitions for a single Timeline element	

6.3.6 Sequence Diagram

A Sequence diagram is a structured representation of behavior as a series of sequential steps over time.

Use To:

- Depict work flow, message passing and how elements in general cooperate over time to achieve a result
- Capture the flow of information and responsibility throughout the system, early in analysis; messages between elements eventually become method calls in the Class model
- Make explanatory models for Use Case scenarios; by creating a Sequence diagram with an Actor and elements involved in the Use Case, you can model the sequence of steps the user and the system undertake to complete the required tasks

Construction:

- Each sequence element is arranged in a horizontal sequence, with messages passing back and

forward between elements

- Messages on a Sequence diagram can be of several types; the Messages can also be configured to reflect the operations and properties of the source and target elements (see the Notes in the Message topic)
- An Actor element can be used to represent the user initiating the flow of events
- Stereotyped elements, such as Boundary, Control and Entity, can be used to illustrate screens, controllers and database items, respectively
- Each element has a dashed stem called a Lifeline, where that element exists and potentially takes part in the interactions

To toggle the numbering of messages on a Sequence diagram, select or deselect the **Show Sequence Numbering** checkbox on the Options dialog.

Example Diagram: [Example Sequence Diagram](#)^[851]

Tools:

Select Sequence diagram elements and connectors from the [Interaction pages](#)^[556] of the Toolbox. Click on the following elements and connectors for more information.

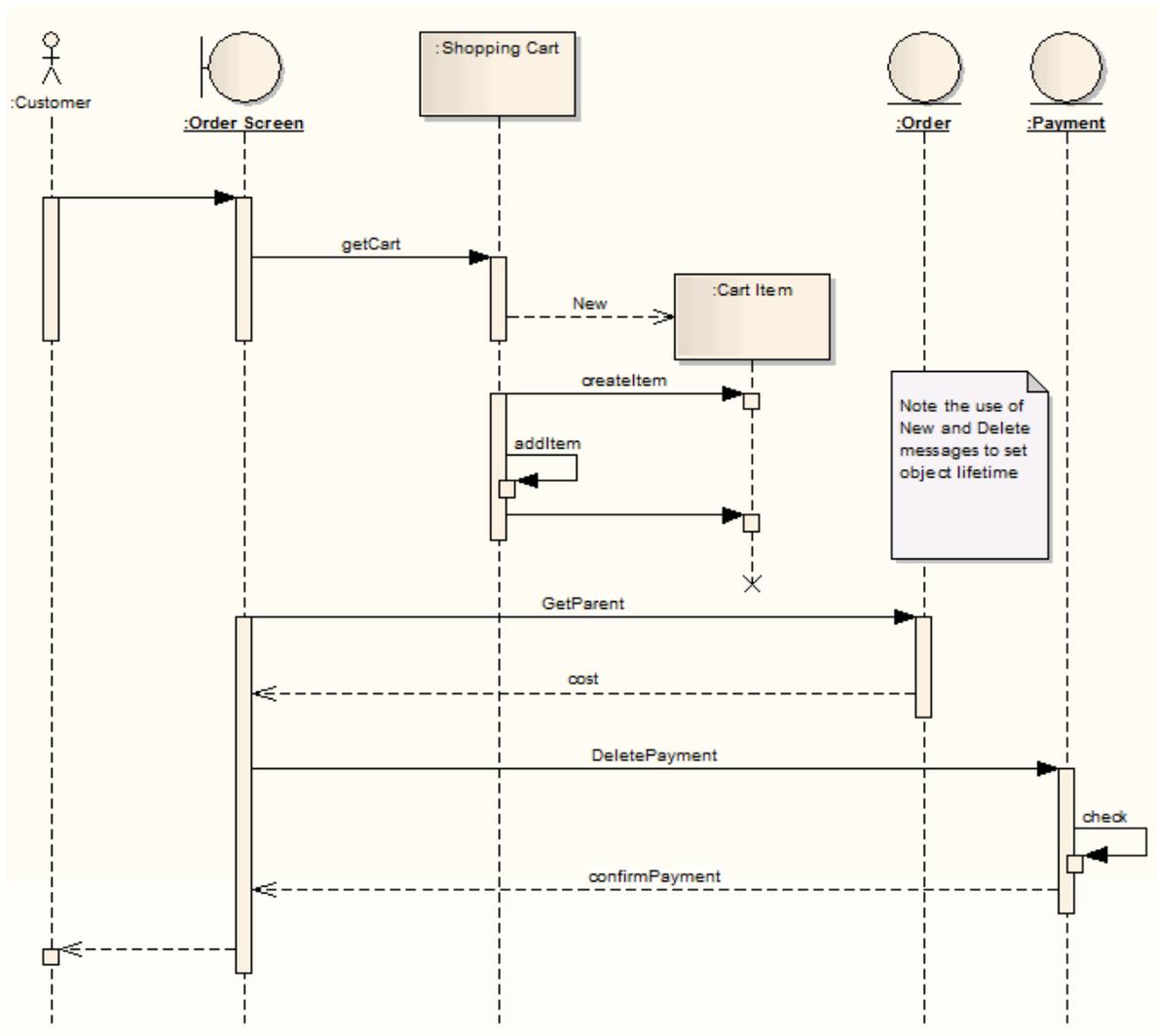
Sequence Diagram Elements	Sequence Diagram Connectors
 Actor	 Message
 Lifeline	 Self-Message
 Boundary	 Recursion
 Control	 Call
 Entity	
 Fragment	
 Endpoint	
 Diagram Gate	
 State/Continuation	

Learn More:

- [Denote the Lifecycle of an Element](#)^[851]
- [Layout of Sequence Diagrams](#)^[852]
- [Sequence Element Activation](#)^[855]
- [Lifeline Activation Levels](#)^[856]
- [Message Label Visibility](#)^[858]
- [Change the Top Margin](#)^[858]
- [Change the Timing Details](#)^[993]
- [Business Modeling/Interaction](#)^[1197]
- [Sequence Diagrams and Version Control](#)^[854]
- [Show Sequence Numbering](#)^[437]

6.3.6.1 Example Sequence Diagram

The following example Sequence diagram demonstrates several different elements:

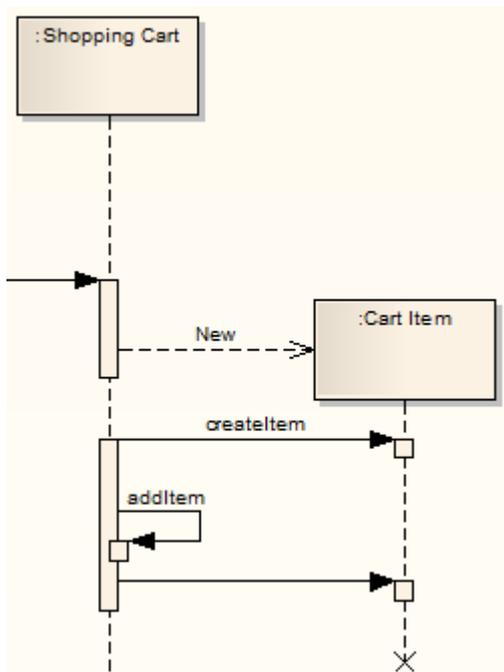


6.3.6.2 Denote Lifecycle of an Element

To capture element lifetimes using messages that are denoted as New or Delete message types, follow the steps below

Step	Action
1	Double-click on a message within a Sequence diagram to display the Message Properties dialog.
2	In the Lifecycle field, click on the drop-down arrow and select New or Delete .
3	Click on the OK button to save the changes.

Example Diagram: The example below shows two elements that have specific creation and deletion times

**Notes:**

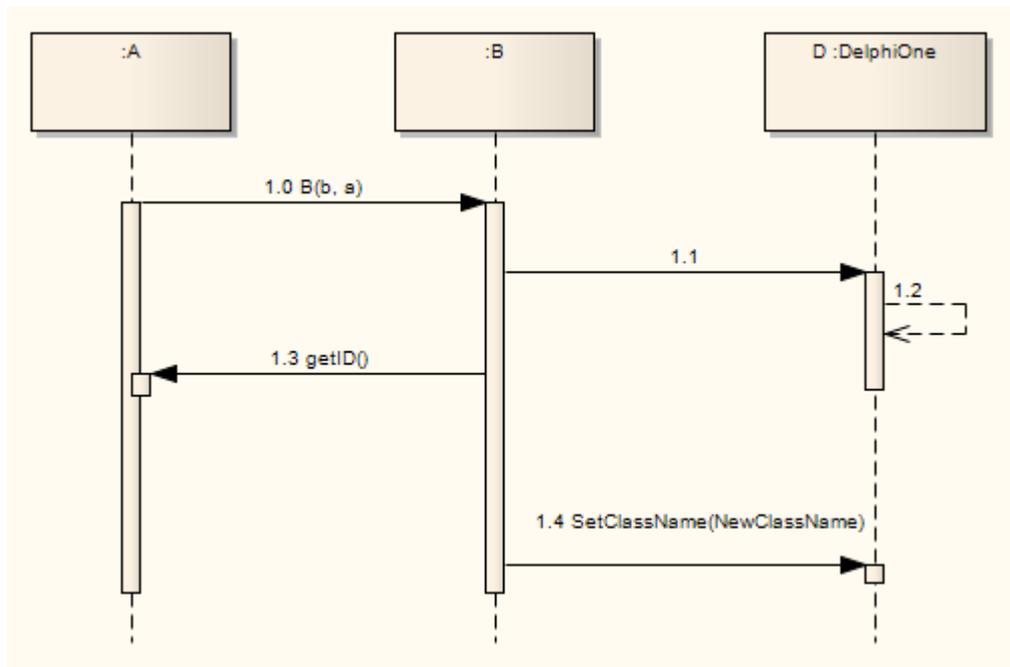
- To show the termination **X** on the lifeline in the example diagram, you must switch on garbage collection: **Tools | Options | Diagram > Sequence: Garbage Collect.**

6.3.6.3 Layout of Sequence Diagrams

To offset the vertical height of sequence messages to get an attractive and effective layout, follow the steps below

Step	Action
1	Select the appropriate message in a Sequence diagram
2	Use the mouse to drag the message up or down as required As you drag a message up or down a lifeline, any messages or fragments below that message are shifted up or down the same amount; however, be aware that if you drag up or down past the next or previous message, Enterprise Architect interprets that as the requirement to swap positions, rather than simply offset a message position

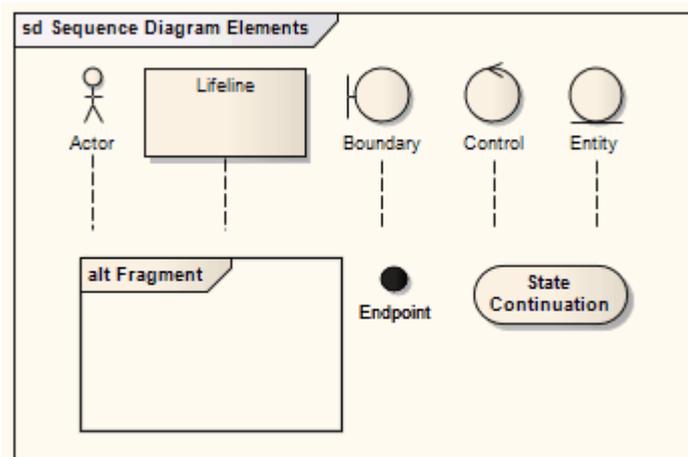
Example Diagram: The example below shows an economical use of space in a Sequence diagram



6.3.6.4 Sequence Elements

The example below shows some possible elements of Sequence diagrams and their stereotyped display.

Element	Description	See Also
Actor	An instance of an actor at runtime; this can be depicted either as the human figure or in rectangle notation.	Actor ^[877]
Lifeline	An Object element with the stereotype Lifeline.	Lifeline ^[910]
Boundary	Represents a user interface screen or input/output device.	Boundary ^[1288]
Entity	A persistent element - typically implemented as a database table or element.	Control ^[1289]
Control	The active component that controls what work gets done, when and how.	Entity ^[1290]



6.3.6.5 Sequence Diagrams and Version Control

You might create Sequence diagrams that use elements from other packages as the Lifelines within the diagram. In such cases, the diagrams could be corrupted when the element packages are checked in and out under version control. This is because during checkout the elements are first deleted from the model and then re-imported, and although they are reinstated in the diagrams, any Messages connecting them are not.

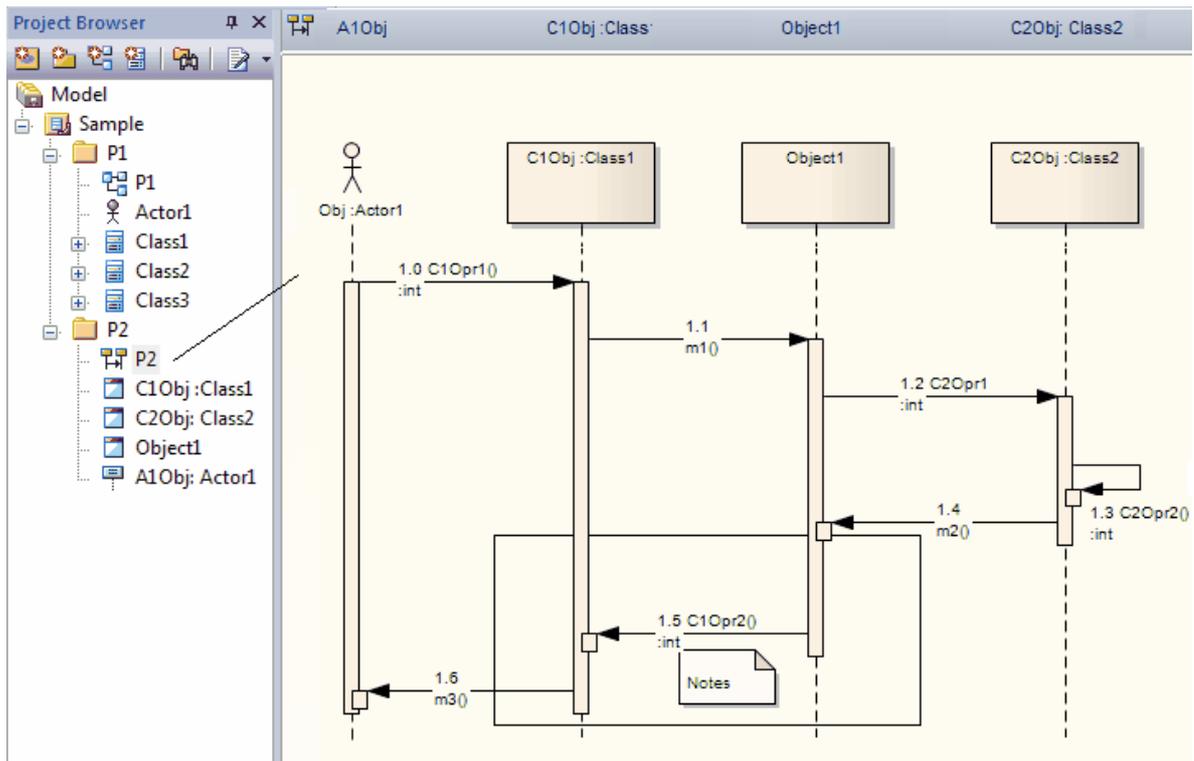
So, if the diagram and its elements reside in different packages, a round-trip of the element package through version control might damage the Sequence diagram.

The solution is to drag-and-drop each Class onto the Sequence diagram as an *object* - when you drop the Class onto the Sequence diagram, in the **Paste Element** dialog select the **as Instance of Element (Object)** option. This creates a new object in the *diagram's* parent package, based on the selected Class element. You then create the Messages between the objects.

Therefore, to ensure that a Sequence diagram is not damaged by round-trips of other packages through version control, remember that:

- The Lifelines must be objects (even though Enterprise Architect allows you to drop elements as Lifelines onto a Sequence diagram, it is not a strictly UML compliant construct)
- The Lifelines must be in the same package as the diagram.

The following illustration shows the **Project Browser** with two packages: *P1*, containing the elements, and *P2*, containing a Sequence diagram that uses those elements. The diagram itself is also shown.



This diagram is not damaged, because all the Lifelines are objects and these objects reside in the same package as the Sequence diagram.

6.3.6.6 Sequence Element Activation

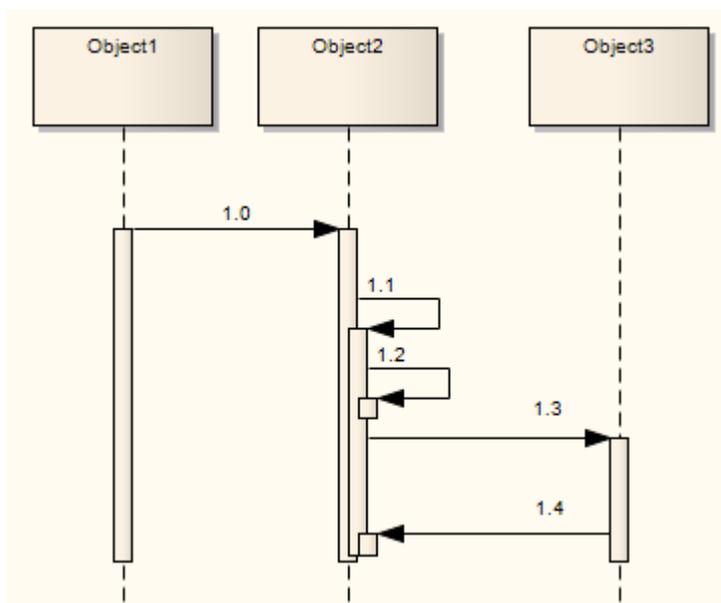
Sequence elements in a Sequence diagram have Activation rectangles drawn along their lifelines. These rectangles describe the time the element is active during the overall period of processing. This visual representation can be suppressed by right-clicking the Sequence diagram, and selecting the **Suppress Activations** context menu option.

In general, Enterprise Architect calculates the period of activation for you, but in some cases you might want to fine tune the rectangle length. There are several context menu options on a sequence message that you can use to accomplish this. To access the following context menu, right-click on the message and select the **Activations** context menu option.

Command	Description
Start New Message Group	Starts off a new round of processing in the current diagram. This enables you to describe more than one processing scenario in a single diagram.
Extend Source Activation Down	Forces an element to stay active beyond the normal processing period. This could be used to express an element that continues its own processing concurrently with other processes.
Extend Source Activation Up	Forces an element's activation upwards.
End Source Activation	Truncates the activation of the source element after the current message. This is useful for expressing an asynchronous message after which the source element becomes idle.
End Target Activation	Ends a Forced Activation started by the Extend Source Activation

	options.
Raise Activation Level	Displays on the context menu only where its use is appropriate. For example, after a self-message the next message starts by default at a lower activation level but the Raise Activation Level command displays on the context menu to enable you to raise its level.
Lower Activation Level	Displays on the context menu only where its use is appropriate.

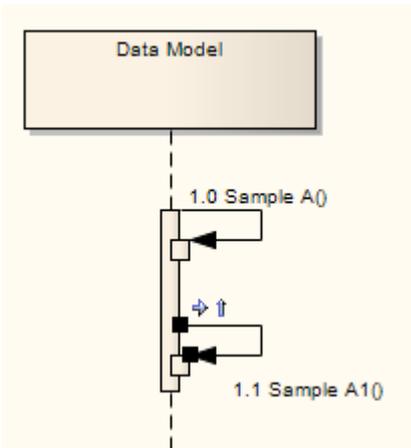
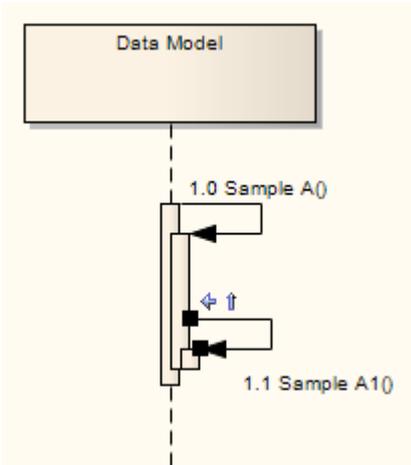
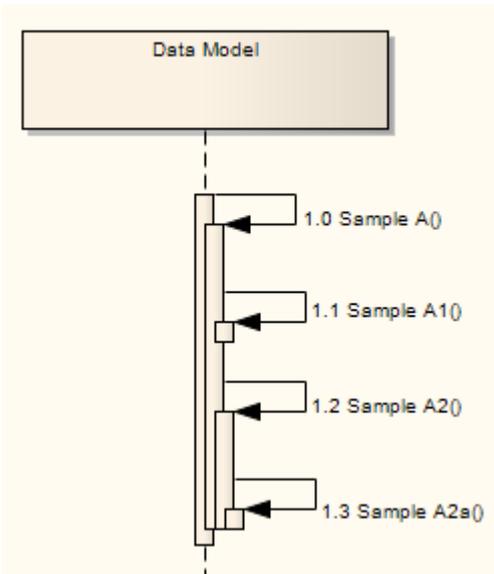
A more convenient way to change activation levels is directly on the diagram. Whenever appropriate, left and/or right arrows display on specific connectors. In the following diagram, see connector 1.3. Click on the arrow to raise or lower the activation level.



6.3.6.7 Lifeline Activation Levels

Complicated processing systems can be easily negotiated and reflected in Sequence diagrams, by adding activation layers on a single lifeline.

Example	Description
---------	-------------

 <p>The diagram shows a 'Data Model' box at the top. Below it is a vertical dashed line representing a lifeline. Two activation bars are shown: the top one is labeled '1.0 Sample A()' and the bottom one is labeled '1.1 Sample A1()'. A message arrow points from the top bar to the bottom bar. A blue arrow points to the connector between these two messages.</p>	<p>A Class invokes the method Sample A, which in turn calls Sample A1.</p> <p>To produce the arrangement in the diagram, select the More tools Interaction menu option, click on the Self-message icon in the Interaction Relationships panel and then click on the lifeline.</p>
 <p>This diagram is identical to the first one, but the blue arrow now points to a small upward-pointing arrow on the connector between the two messages, indicating the 'raise' operation.</p>	<p>In order to raise the Activation level of Sample A1, click on the raise arrow of the selected connector. The lifeline now visually depicts that method Sample A1 is called during the processing of Sample A.</p>
 <p>The diagram shows a single lifeline with four activation bars. The top bar is '1.0 Sample A()', followed by '1.1 Sample A1()', '1.2 Sample A2()', and '1.3 Sample A2a()' at the bottom. Each bar is connected to the next one below it by a message arrow, showing a sequence of self-calls.</p>	<p>In the example below, a few more self-messages have been added. The message Sample A2a is called from Sample A2 which in turn is called from Sample A (not Sample A1). Sample A1 is called from Sample A.</p>

6.3.6.8 Sequence Message Label Visibility

To hide and show the labels used in Sequence messages, follow the steps below

Step	Action
1	Right-click on the message within the Sequence diagram The message context menu displays
2	Select the Set Label Visibility menu option The Label Visibility dialog displays
3	Select or clear the checkbox against each message label to display or hide, respectively
4	Click on the OK button to save the settings

6.3.6.9 Change the Top Margin

In order to change the top margin of a Sequence diagram from the default 50 units, right-click on the diagram to display the context menu and select the **Set Top Margin** menu option. You can set the top margin to any value between 30 and 250 units.

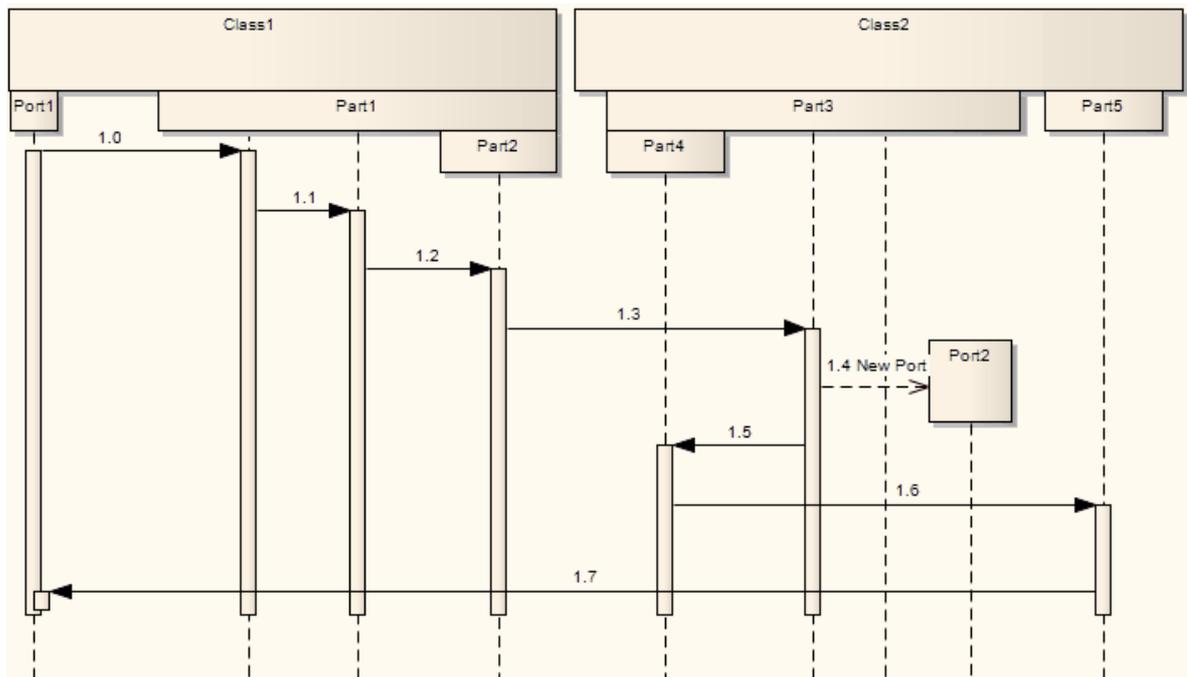
6.3.6.10 Inline Sequence Elements

It is possible to represent Part and Port elements on a Sequence diagram. Child Parts and Ports appear as inline sequence elements under their parent Class sequence element.

How to:

To represent Part and Port elements on a Sequence diagram, follow the steps below

Step	Action
1	Right-click on the sequence elements containing the child Ports or Parts, to display the context menu
2	Select the Embedded Elements Embedded Elements menu option
3	Select the checkbox against each Part or Port to show, and click on the Close button



6.3.7 Communication Diagram

A Communication diagram is a diagram that shows the interactions between elements at run-time in much the same manner as a Sequence diagram. However, Communication diagrams are used to visualize inter-object relationships, while Sequence diagrams are more effective at visualizing processing over time. Communication diagrams were known as Collaboration diagrams in UML 1.

Communication diagrams employ ordered, labeled associations to illustrate processing. Numbering is important to indicate the order and nesting of processing. A numbering scheme could be:

1
 1.1
 1.1.1
 1.1.2
 1.2, and so on.

A new number segment begins for a new layer of processing, and would be equivalent to a method invocation.

Example Diagram: [Example Communication Diagram](#) 

Tools: Select Communication diagram elements and connectors from the Communication pages of the **Toolbox**. Click on the following elements and connectors for more information.

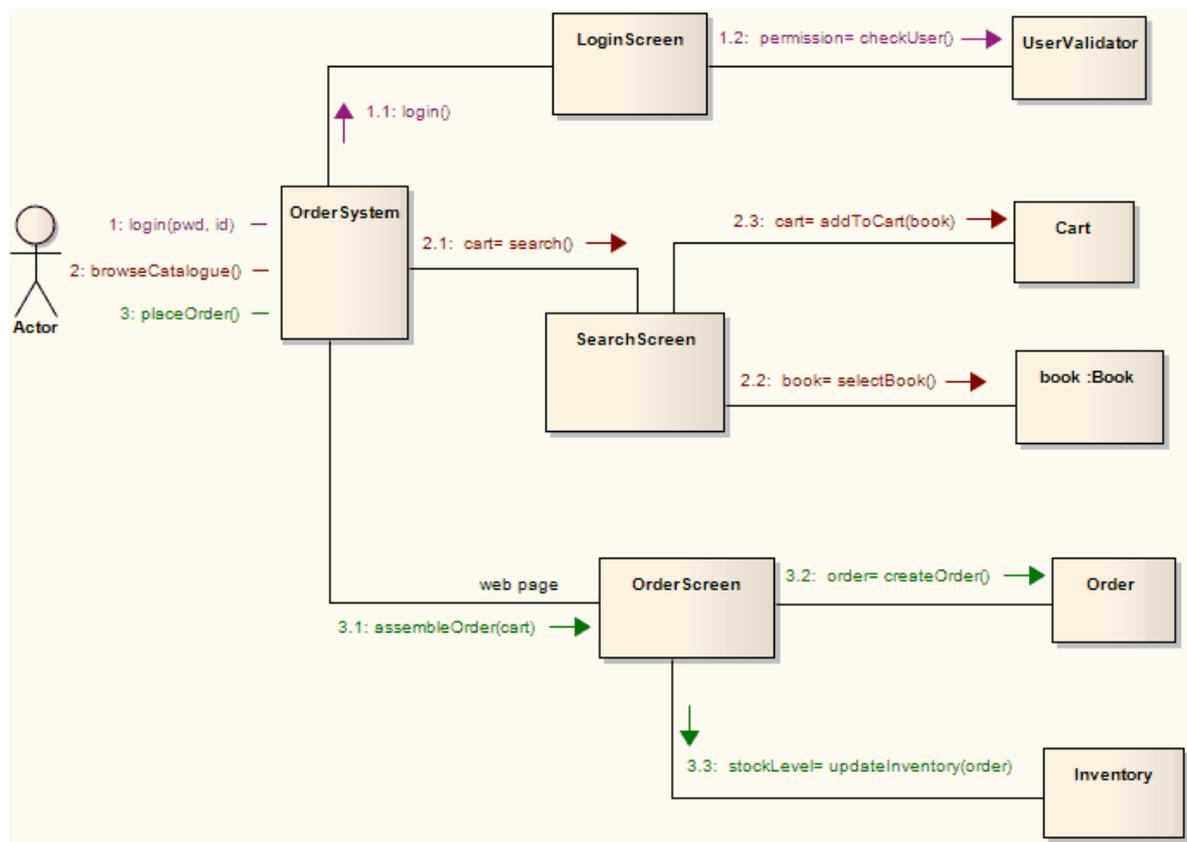
Communication Diagram Elements	Communication Diagram Connectors
 Actor	 Associate
 Object	 Nesting
 Boundary	 Realize
 Control	
 Entity	
 Package	

Learn More:

- [Communication Diagrams in Color](#)^[86]

6.3.7.1 Example Communication Diagram

This example illustrates a Communication diagram among cooperating object instances. Note the use of message levels to capture related flows, and the different colors of the messages.

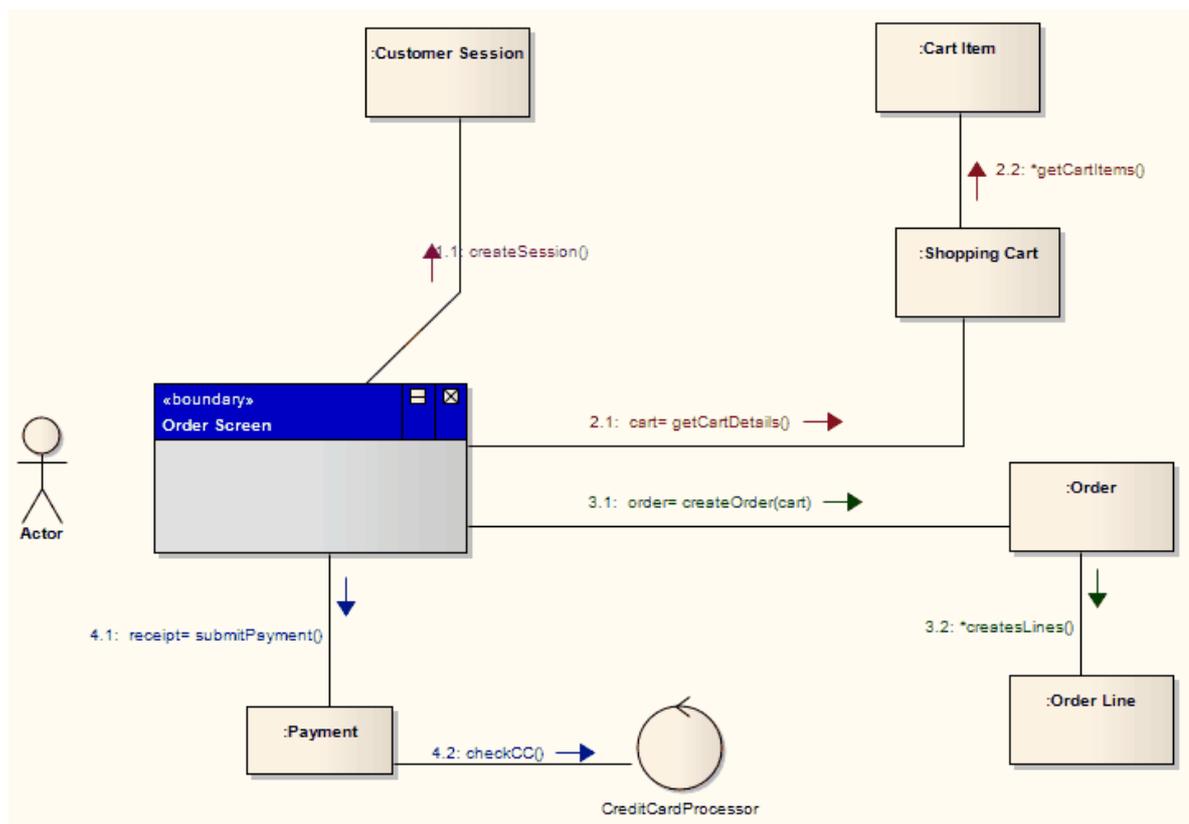


6.3.7.2 Communication Diagrams in Color

Enterprise Architect enables you to highlight particular message flows in a Communication diagram using different colors for each message set.

To highlight the colors in a Communication diagram, follow the steps below:

Step	Action
1	Select the Tools Options Communication Colors menu option The Communication Message Coloring page of the Options dialog displays
2	Select the Use Communication Color checkbox
3	Click on the drop-down arrow of each Message n field, and select the required color for each message group
4	Click on the Close button On your Communication diagram, each sequence group of messages displays in a different color, as shown below



6.3.8 Interaction Overview Diagram

Interaction Overview diagrams visualize the cooperation between other interaction diagrams to illustrate a control flow serving an encompassing purpose. As Interaction Overview diagrams are a variant of Activity diagrams, most of the diagram notation is the same, as is the process of constructing the diagram. Decision points, Forks, Joins, Start points and End points are the same. Instead of Activity elements, however, rectangular elements are used. There are two types of these elements:

- Interaction elements display an inline Interaction diagram, which can be any one of the four types

- Interaction Occurrence elements are references to an existing Interaction diagram: they are visually represented by a frame, with **ref** in the frame's title space; the diagram name is indicated in the frame contents.

To create an Interaction Occurrence, simply drag an Interaction diagram from the **Project Browser** onto your Interaction Overview diagram. The **ref** frame displays, encapsulating an instance of the Interaction diagram.

Example Diagram: [Example Interaction Overview Diagram](#)^[862]

Tools: Select Interaction Overview diagram elements and connectors from the Activity pages of the Toolbox. Click on the following elements and connectors for more information.

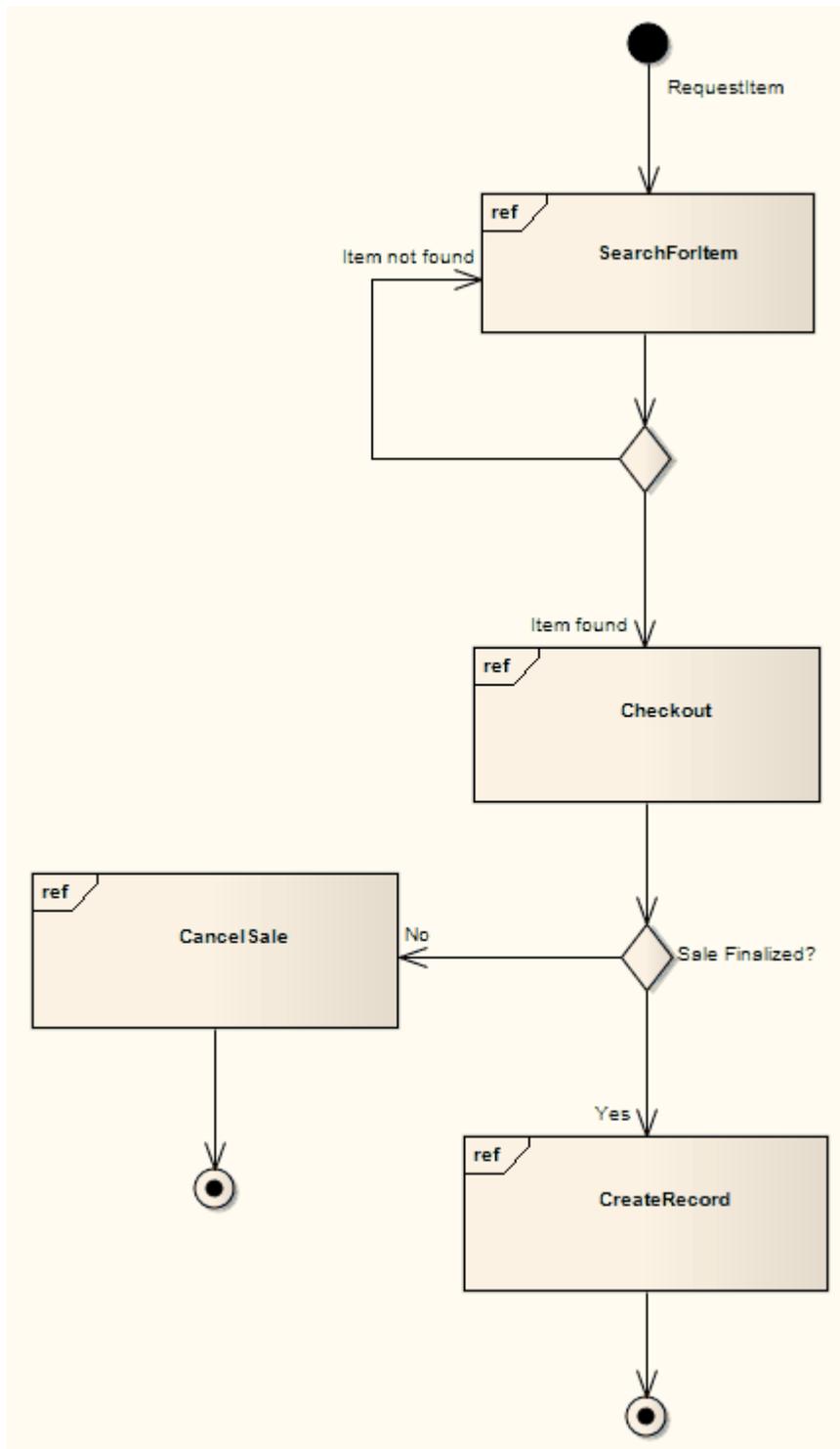
Interaction Overview Diagram Elements	Interaction Overview Diagram Connectors
 Partition	 Control Flow
 Decision	 Object Flow
 Send	 Interrupt Flow
 Receive	
 Synch	
 Initial	
 Final	
 Flow Final	
 Region	
 Exception	
 Merge	
 Fork/Join	
 Fork/Join	

Learn More:

- [Activity diagrams](#)^[811]
- [Interaction](#)^[904]
- [Interaction Occurrence](#)^[906]

6.3.8.1 Example Interaction Overview Diagram

This diagram depicts a sample sale process, shown in an Interaction Overview diagram, with sub-processes abstracted within Interaction Occurrences. The diagram appears very similar to an Activity diagram, and is conceptualized the same way; as the flow moves into an interaction, the respective interaction's process must be followed before the Interaction Overview's flow can advance.



6.4 UML Elements

Models in UML are constructed from elements such as *Classes*, *Objects*, *Interfaces*, *Use Cases*, *Components* and *Nodes*, each of which has a different purpose, different rules and different notation. Model elements are used at different stages of the design process for different purposes. For example,

- During early analysis, Use Cases, Activities, Business Processes, Objects and Collaborations are used to capture the problem domain
- During elaboration, Sequence diagrams, Objects, Classes and State Machines are used to refine the system specification
- Components and Nodes are used to model larger parts of the system as well as the physical entities that are created and deployed into a production environment.

Topics:

Topic	Detail	See also
Behavioral Diagram Elements	Behavioral diagrams depict the behavioral features of a system or business process. Elements that can appear on Behavioral diagrams include Activity, Interaction, Lifeline, State Machine and Use Case.	Behavioral Diagram Elements ^[864]
Structural Diagram Elements	Structural diagrams depict the structural elements composing a system or function. Elements that can appear on Structural diagrams include Class, Component, Interface, Node and Package.	Structural Diagram Elements ^[940]

6.4.1 Behavioral Diagram Elements

The following elements are used in UML Behavioral Diagrams. For more information on using each element, click on the element name in this list:

- [Action](#) ^[865], [Activity](#) ^[873], [Actor](#) ^[877]
- [Central Buffer Node](#) ^[878], [Choice](#) ^[879], [Collaboration](#) ^[945], [Collaboration Use](#) ^[946], [Combined Fragment](#) ^[880]
- [Datastore](#) ^[885], [Decision](#) ^[886], [Diagram Frame](#) ^[888], [Diagram Gate](#) ^[889]
- [Endpoint](#) ^[890], [Entry Point](#) ^[891], [Exception](#) ^[892], [Expansion Region](#) ^[893], [Exit Point](#) ^[895]
- [Final](#) ^[895], [Flow Final](#) ^[897], [Fork](#) ^[898]
- [History](#) ^[902]
- [Initial](#) ^[903], [Interaction](#) ^[904], [Interaction Occurrence](#) ^[906], [Interruptible Activity Region](#) ^[907]
- [Join](#) ^[898], [Junction](#) ^[908]
- [Lifeline](#) ^[910]
- [Merge](#) ^[910], [Message Endpoint](#) ^[911], [Message Label](#) ^[912]
- [Note](#) ^[913]
- [Object](#) ^[957]
- [Package](#) ^[960], [Partition](#) ^[914]
- [Receive](#) ^[915], [Region](#) ^[916]
- [Send](#) ^[917], [State](#) ^[917], [State/Continuation](#) ^[920], [State Lifeline](#) ^[923], [State Machine](#) ^[925], [Structured Activity](#) ^[926], [Synch](#) ^[930], [System Boundary](#) ^[931]
- [Terminate](#) ^[933], [Trigger](#) ^[934]
- [Use Case](#) ^[935]
- [Value Lifeline](#) ^[936]

Learn More:

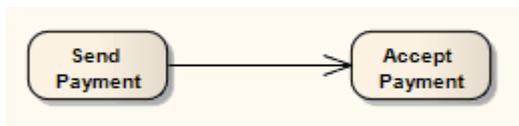
- [Behavioral Diagrams](#) ^[811]

6.4.1.1 Action



Description:

An Action element describes a basic process or transformation that occurs within a system. It is the basic functional unit within an Activity diagram. Actions can be thought of as children of Activities. Both represent processes, but Activities can contain multiple steps or decomposable processes, each of which can be embodied in an Action. An Action cannot be further broken down or decomposed.



An Action can be further defined with pre-condition and post-condition notes, and certain properties can be graphically depicted on the Action (Enterprise Architect prompts you to define the type of Action you are creating when you first drag the **Action** icon from the Toolbox). The data values passed out of and into an Action can be represented by Action Pins. For a named Action (that is, other than a basic Action) you can also assign Action Pins to represent specific properties.

For a basic (Atomic) Action, you can define the effect of the Action using the Effect tab of the element Properties dialog, and select to display the effect on the diagram.

An Action can also be depicted as an Expansion Node to indicate that the Action comprises an Expansion Region.

Toolbox Icon:



Learn More:

- [Action Notation](#) ^[866]
- [Action Expansion Node](#) ^[868]
- [Action Pin](#) ^[869]
- [Assign Action Pins](#) ^[870]
- [Local Pre/Post Conditions](#) ^[871]
- [Class Operations in Activity Diagrams](#) ^[872]
- [Activity Diagram](#) ^[811]
- [Activity Element](#) ^[873]
- [Expansion Region](#) ^[893]

OMG UML Specification:

The OMG UML specification (UML Superstructure Specification, v2.1.1, p. 241) states:

An action is a named element that is the fundamental unit of executable functionality. The execution of an action represents some transformation or processing in the modeled system, be it a computer system or

otherwise.

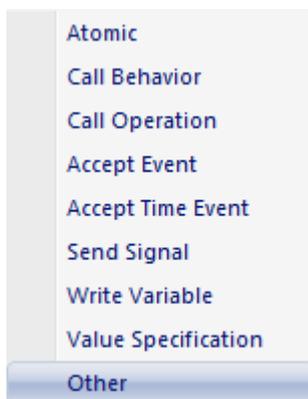
The OMG UML specification (UML Superstructure Specification, v2.1.1, p. 313) also states:

An action may have sets of incoming and outgoing activity edges that specify control flow and data flow from and to other nodes. An action will not begin execution until all of its input conditions are satisfied. The completion of the execution of an action may enable the execution of a set of successor nodes and actions that take their inputs from the outputs of the action.

6.4.1.1.1 Action Notation

Description:

When you drag the Action icon from the Activity page of the Toolbox onto your diagram, a selection list displays showing the commonest types of Action to create. (If this list does not display, press **(Ctrl)** as you drag the icon.)

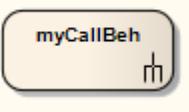


When you click on one of the specific types, that type of Action element displays on diagram. If you click on the **Other** option, the New Action dialog displays. You can again select to create a normal (Atomic) Action element, a CallOperation or a CallBehavior, or you can select the **Other** radio button and click on the drop-down arrow in the blank field to select the Action type from an extensive list.

If you later decide that the Action type is not appropriate, you can change it by right-clicking on the Action and selecting the **Properties** context menu option, and then selecting the **Advanced** page of the Properties dialog. Set the Action type by selecting a value from the **Kind** drop-down list. For a Value Specification Action, you also set the value on this page.

Action Kind Notation

Action Kind	Notation	See Also
AcceptEvent		AcceptEvent Actions ⁸⁶⁷
AcceptEventTimer		

Action Kind	Notation	See Also
CallBehavior		
CallOperation		
SendSignal		SendSignal Action & BroadcastSignal Action ⁸⁶⁷

AcceptEvent Actions:

For an *Accept Event* Action element, the Properties dialog contains a Triggers tab on which you define one or more triggers to denote the type of events accepted by the Action, as defined in the following table:

Field	Usage	See also
Name	Specify the name of the trigger	
Type	<p>Specify the type of trigger: Call, Change, Signal or Time</p> <ul style="list-style-type: none"> Call - specifies that the event is a CallEvent, which sends a message to the associated object by invoking an operation Change - specifies that the event is a ChangeEvent, which indicates that the transition is the result of a change in value of an attribute Signal - specifies that the event is a SignalEvent, which corresponds to the receipt of an asynchronous signal instance Time - corresponds to a TimeEvent; which specifies a moment in time <p>Code generation for State Machines currently supports Change and Time trigger events only, and expects a specification value</p>	
Specification	Specify the event instigating the Transition	

SendSignal Action & BroadcastSignal Action:

For a *SendSignal* or *BroadcastSignal* Action element, you can model the signal to be sent and the associated arguments to be conveyed, using the Signal tab of the element Properties dialog.

To model the signal to be sent and the associated arguments to be conveyed, follow the steps below:

Step	Action
1	In the Signal field, click on the (...) button and select the required signal from the Select Signal

Step	Action
	dialog
2	In the Attribute field, click on the drop-down arrow and select the attribute (as previously created in the Signal element) with which the arguments are to be associated
3	In the Value field, type the appropriate value for the attribute
4	Identify the arguments (as ActionPins) for the Signal; click on the Add button under the Arguments panel, and select the appropriate Pins from the Select Pin dialog To assign more than one Pin, press (Ctrl) whilst you select each Pin
5	Click on the Save button

Structural Feature Actions:

Enterprise Architect supports the following types of *Structural Feature* Actions:

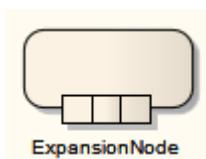
- AddStructuralFeatureValue
- ClearStructuralFeature
- ReadStructuralFeature
- RemoveStructuralFeatureValue
- WriteStructuralFeature

These actions can take Ports, Parts or Attributes as the target structural feature.

To set the appropriate target structure feature, follow the steps below:

Step	Action
1	Right-click on the Action element in the diagram, and select the Advanced Set Structural Feature context menu option The Set Structural Feature dialog displays
2	To locate the structural feature, click on the Add button The Select Property dialog displays (a variant of the Select <Item> dialog)
3	Browse or search for the appropriate structural feature, and double-click on it The feature name and location displays in the structuralFeature field of the Set Structural Feature dialog
4	Click on the OK button to save the setting

6.4.1.1.2 Action Expansion Node



Description

Representing an Action as an *Expansion Node* is a shorthand notation to indicate that the Action comprises an Expansion Region.

To specify an Action as an Expansion Node, right-click on the Action to display the context menu and select the **Embedded Elements | Add Expansion Node** menu option. After designating an Action as an Expansion Node, you can modify or delete it using the **Embedded Elements | Embedded Elements** menu option.

See Also:

- [Action](#) ^[865]
- [Expansion Region](#) ^[893]

6.4.1.1.3 Action Pin



Description

An *Action Pin* is used to define the data values passed out of and into an Action. An *input pin* provides values to the Action, whereas an *output pin* contains the results from that Action.

Action Pins are used below to connect two Actions:



Action Pins can be further characterized as defining exception parameters, streams, or states. Associating a state with a Pin defines the state of input or output values. For instance, the Pin could be called *Orders*, but the state could be *Validated* or *Canceled*.

To *add* an Action Pin to an Action, right-click on the Action to display the context menu and select the **Embedded Elements | Add Action Pin** menu option. (You can also assign Action Pins, to define specific properties of the Action.)

The Properties dialog of an Action Pin has a Pin tab on which you define the specific actions of the Pin.

A Pin serves as an argument for Call Behavior Actions and Call Operation Actions. When an Action is associated with a valid behavior in the model, the associated behavior's parameters are listed in the **Parameter** field drop-down list to facilitate a one-to-one mapping between the argument and the parameter. The fields in the Argument panel are enabled only for Pins belonging to Call Actions, and only when the Action is associated with a valid behavior with valid parameters.

You can also change certain properties of an Action Pin on the Advanced page of the element Properties dialog: right-click on the Pin and select the **Properties** context menu option. The *objectState* and *kind* properties can be set.

See Also:

- [Action](#) ^[865]
- [Assign Action Pins](#) ^[870]

6.4.1.1.4 Assign Action Pins

Apart from *adding* Action Pins to any Action, you can *assign* specialized input or output Action Pins to Actions that have a specific type (that is, those that are not Basic or Atomic Actions). These input/output Pins signify various *properties* of the Action - they are not visible as structures on the diagram unless they have previously been added, but are listed in the Project Browser as properties of the Action.

You can only assign Pins that have already been added or assigned to the Action, or that are being created specifically to be assigned to the Action.

How to:

To assign Pins to an Action, follow the steps below:

Step	Action	See also
1	<p>Right-click on the Action in the diagram, and select the Advanced Assign Action Pins context menu option</p> <p>The Assign Action Pins to <ActionName> dialog displays</p> <p>The format of this dialog depends on the type of Action: for a:</p> <ul style="list-style-type: none"> • <i>SendObject</i> Action the dialog has two fields (request and target) • <i>TestIdentity</i> Action, three • <i>CallBehavior</i> Action, one (result) <p>The fields are populated in exactly the same way</p>	
2	<p>The mandatory number and type of Pins are automatically selected (if they exist) or created</p> <p>To change or add a Pin in a field, click on the corresponding Add button</p> <p>The Select Pins dialog displays, showing the selected Action and listing all the input Pins currently owned by the Action</p>	Instance Classifier [690]
3	<p>Double-click on one of the Pins (or, depending on the multiplicity of the Pin, (Ctrl) +click on several Pins)</p> <p>Alternatively, if no suitable Pin exists, click on the Add New button and then click on the newly-created Pin</p> <p>The selected Pin is identified in the field on the Assign Action Pins to <ActionName> dialog</p>	
4	Click on the OK button	

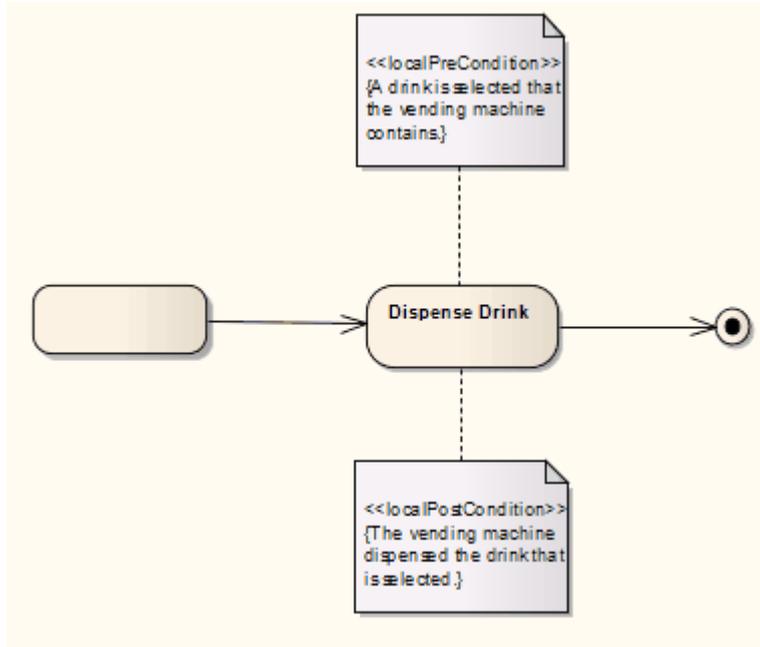
To check the exact location of an assigned Action Pin, you can right-click on the Pin name in the dialog and select the **Find in Project Browser** context menu option.

Learn More:

- [Action Pin](#) [869]

6.4.1.1.5 Local Pre/Post Conditions

Actions can be further defined with pre-condition and post-condition notes, which constrain an Action's entry and exit. These notes can be added to an Action as defined below.



How to:

To attach a constraint to an Action, follow the steps below:

Step	Action
1	Right-click on the Action The context menu displays
2	Select the Add Constraint menu option A <i>Note</i> is created on the diagram, connected to the Action
3	Right-click on the <i>Note</i> The context menu displays
4	Select the View Properties menu option The Constraint dialog displays
5	In the Constraint Type field, click on the drop-down arrow and select the required constraint type
6	In the Constraint field, type the text for the constraint
7	Click on the OK button to save the constraint

Learn More:

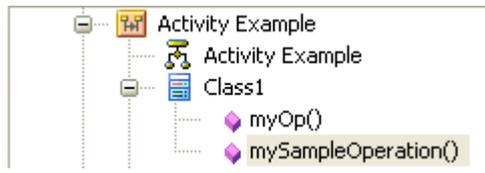
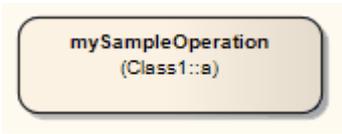
- [Action](#) [865]

6.4.1.1.6 Class Operations in Activity Diagrams

Operations from Classes can be displayed on Activity diagrams as Actions. When an operation is shown as an Action, the notation of the Action displays the name of the Class that features the operation.

How to:

To add an operation to an Activity diagram, follow the steps below:

Step	Action	See Also
1	Open an Activity diagram	
2	From the Project Browser open a Class and locate the operation to be added to the Activity diagram	
3	Drag the operation on to the diagram 	
4	When the operation has been added to the Activity diagram the Action displays the name of the Class that features the operation  Right-click on the Action in the diagram to locate, in the Project Browser, the: <ul style="list-style-type: none"> • Behavior classifier (CallBehavior Activity) using the Find Locate Classifier in Project Browser context menu option or • Call operation (CallOperation Activity) using the Find Locate Operation in Project Browser context menu option 	

If it becomes necessary to change the operation that this Action refers to, follow the steps below:

Step	Action	See Also
1	Right-click on the Action The context menu displays	
2	Select the Advanced Set Operation menu option The Set Operation dialog displays	Set Operation dialog ⁶⁹⁴
3	If necessary, in the In Namespace field, select the model that contains the required operation	

Step	Action	See Also
4	Double-click on the required operation. The Action updates to show the new classifier and operation <div style="border: 1px solid black; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"> mySampleOperation (AccountItem::GetName) </div>	

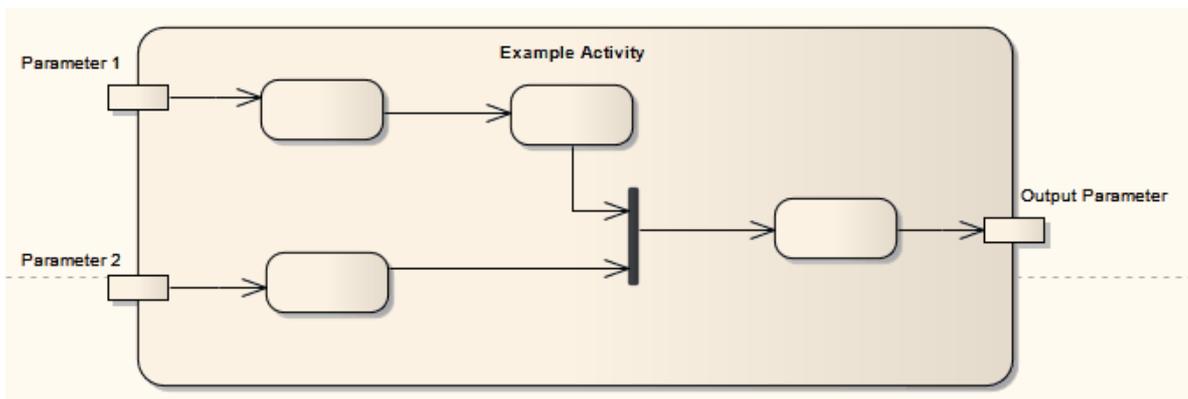
6.4.1.2 Activity



Description

An Activity organizes and specifies the participation of subordinate behaviors, such as sub-Activities or Actions, to reflect the control and data flow of a process. Activities are used in Activity diagrams for various modeling purposes, from procedural-type application development for system design, to business process modeling of organizational structures or work flow.

The following simple diagram of an Activity contains Action elements and includes input parameters and output parameters.



You can define an Activity as a composite element, either during creation or during later edits. When creating a composite Activity element, it is simpler to apply the mechanism for creating Structured Activity elements, which reduces the number of steps to work through. See the Structured Activity topic. If converting an existing Activity element, right-click on the element and select the **Advanced | Make Composite** context menu option.

Certain properties can be graphically depicted on an Activity. The Actions in an Activity can be further organized by Activity Partitions.

Toolbox Icon

Activity

See Also:

- [Activity Notation](#) ^[874]
- [Activity Parameter Nodes](#) ^[875]
- [Activity Partition](#) ^[876]

Learn More:

- [Action](#) ^[865]
- [Activity Diagram](#) ^[817]
- [Composite Element](#) ^[647]
- [Structured Activity](#) ^[926]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 318*) states:

An activity specifies the coordination of executions of subordinate behaviors, using a control and data flow model. The subordinate behaviors coordinated by these models may be initiated because other behaviors in the model finish executing, because objects and data become available, or because events occur external to the flow. The flow of execution is modeled as activity nodes connected by activity edges. A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents. Activity nodes also include flow-of-control constructs, such as synchronization, decision, and concurrency control. Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions. In an object-oriented model, activities are usually invoked indirectly as methods bound to operations that are directly invoked.

Activities may describe procedural computation. In this context, they are the methods corresponding to operations on classes. Activities may be applied to organizational modeling for business process engineering and workflow. In this context, events often originate from inside the system, such as the finishing of a task, but also from outside the system, such as a customer call. Activities can also be used for information system modeling to specify system level processes. Activities may contain actions of various kinds:

- Occurrences of primitive functions, such as arithmetic functions.
- Invocations of behavior, such as activities.
- Communication actions, such as sending of signals.
- Manipulations of objects, such as reading or writing attributes or associations.

Actions have no further decomposition in the activity containing them. However, the execution of a single action may induce the execution of many other actions. For example, a call action invokes an operation that is implemented by an activity containing actions that execute before the call action completes.

6.4.1.2.1 Activity Notation

Certain properties can be graphically depicted on an Activity element, as shown below.



To define these properties, right-click on the Activity and select the **Properties** context menu option, then select the Advanced page of the Properties dialog.

See Also:

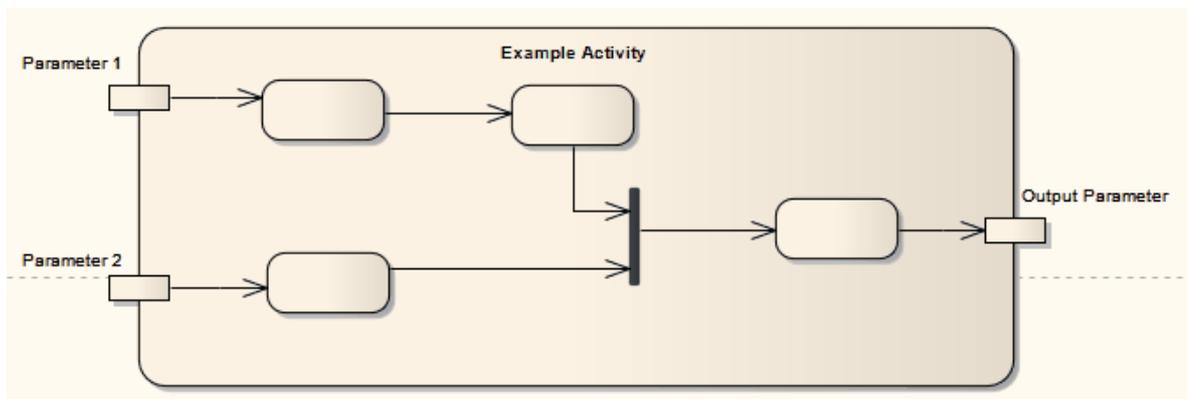
- [Activity](#)^[873]

6.4.1.2.2 Activity Parameter Nodes

Description:

An Activity Parameter Node accepts input to an Activity or provides output from an Activity.

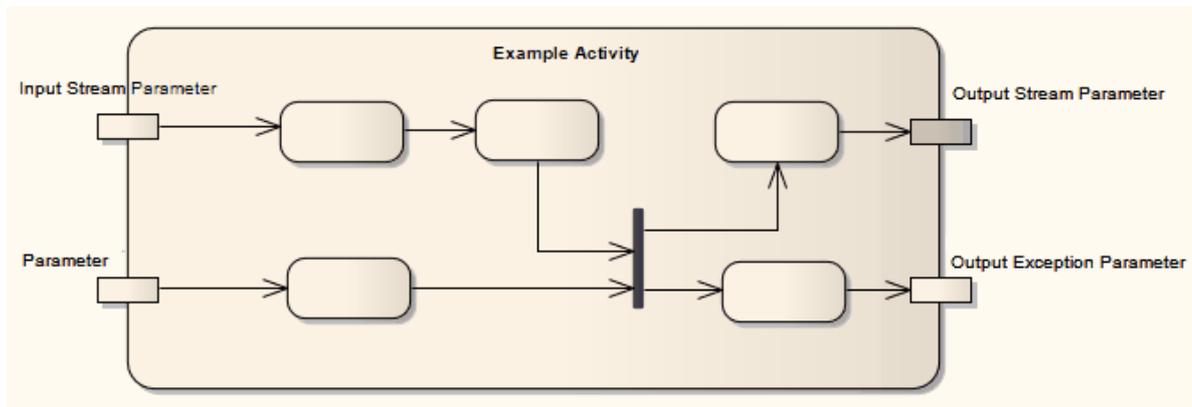
The following example depicts two entry parameters and one output parameter defined for the Activity.



How to:

To define an Activity Parameter Node for an Activity, follow the steps below:

Step	Action
1	Right-click on the element and select the Embedded Elements Add Activity Parameter context menu option
2	The Properties dialog displays, which prompts for the Name and other properties of the embedded element
3	To further define the new Activity Parameter, select the Advanced page of the Properties dialog Similar to characterizing Action Pins, Activity Parameter Nodes also have the isException and isStream options: <ul style="list-style-type: none"> • isException indicates that a parameter can emit a value at the exclusion of other outputs, usually because of some error • isStream indicates whether or not a parameter can accept or post values during the execution of the Activity



Learn More:

- [Activity Element](#)^[873]
- [Action Pin](#)^[869]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 338) states:

An activity parameter node is an object node for inputs and outputs to activities.

... Activity parameter nodes are object nodes at the beginning and end of flows that provide a means to accept inputs to an activity and provide outputs from the activity, through the activity parameters.

Activity parameters inherit support for streaming and exceptions from Parameter.

6.4.1.2.3 Activity Partition

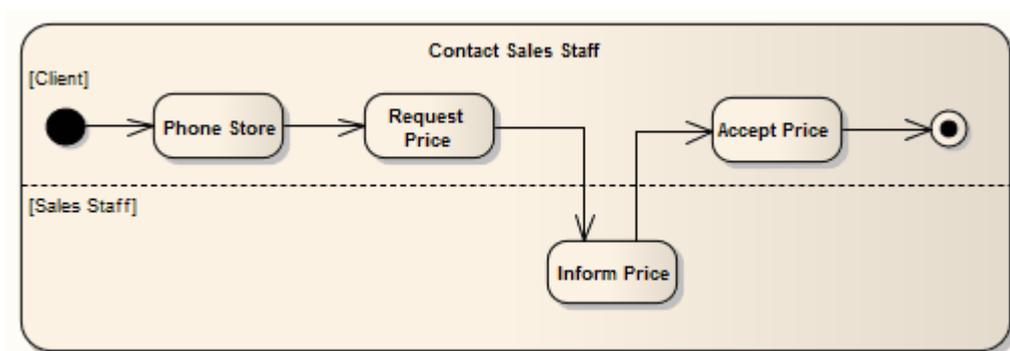
Description

Enterprise Architect supports two types of *Activity Partition*:

- The Activity Partition *feature*, described in this topic, which is used to logically organize an Activity *element*
- The Activity Partition *element*, which is used to logically organize an Activity *diagram*

In effect, these are the same. They partition the Actions of the Activity without affecting the token flow, helping to structure the view or parts of the Activity.

An example of a feature-partitioned Activity is shown below:



How to:

To define Partitions using the feature, follow the steps below:

Step	Action
1	Right-click on the Activity element The context menu displays
2	Select the Advanced Partition Activity menu option The Activity Partitions dialog displays
3	In the Name field, type the name of a partition Click on the Save button
4	Repeat step 3 for each partition to be created

Learn More:

- [Activity Diagram](#)^[811]
- [Activity Element](#)^[873]
- [Activity Partition Element](#)^[914]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) states:

Partitions divide the nodes and edges to constrain and show a view of the contained nodes. Partitions can share contents. They often correspond to organizational units in a business model. They may be used to allocate characteristics or resources among the nodes of an activity.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) also states:

An activity partition is a kind of activity group for identifying actions that have some characteristic in common.

6.4.1.3 Actor**Description**

An Actor is a user of the system; user can mean a human user, a machine, or even another system or subsystem in the model. Anything that interacts with the system from the outside or system boundary is termed an Actor. Actors are typically associated with Use Cases.

Actors can use the system through a graphical user interface, through a batch interface or through some other media. An Actor's interaction with a Use Case is documented in a Use Case scenario, which details the functions a system must provide to satisfy the user requirements.

Actors also represent the role of a user in Sequence Diagrams, where you can display them using rectangle notation. Enterprise Architect supports a stereotyped Actor element for business modeling. The business modeling elements also represent Actors as stereotyped Objects.

Toolbox Icon



Learn More:

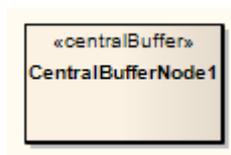
- [Sequence Diagrams](#)^[849]
- [Business Modeling](#)^[1191]
- [Use Case](#)^[935]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 584*) states:

An actor models a type of role played by an entity that interacts with the subject (e.g. by exchanging signals and data), but which is external to the subject. ... Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated Use Cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances.

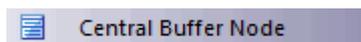
6.4.1.4 Central Buffer Node



Description

A Central Buffer Node is an object node for managing flows from multiple sources and destinations, represented in an Activity diagram. It acts as a buffer for multiple in-flows and out-flows from other object nodes, but does not connect directly to Actions.

Toolbox Icon



Learn More:

- [Activity Diagram](#)^[811]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 352*) states:

A central buffer node is an object node for managing flows from multiple sources and destinations. ... A central buffer node accepts tokens from upstream object nodes and passes them along to downstream object nodes.

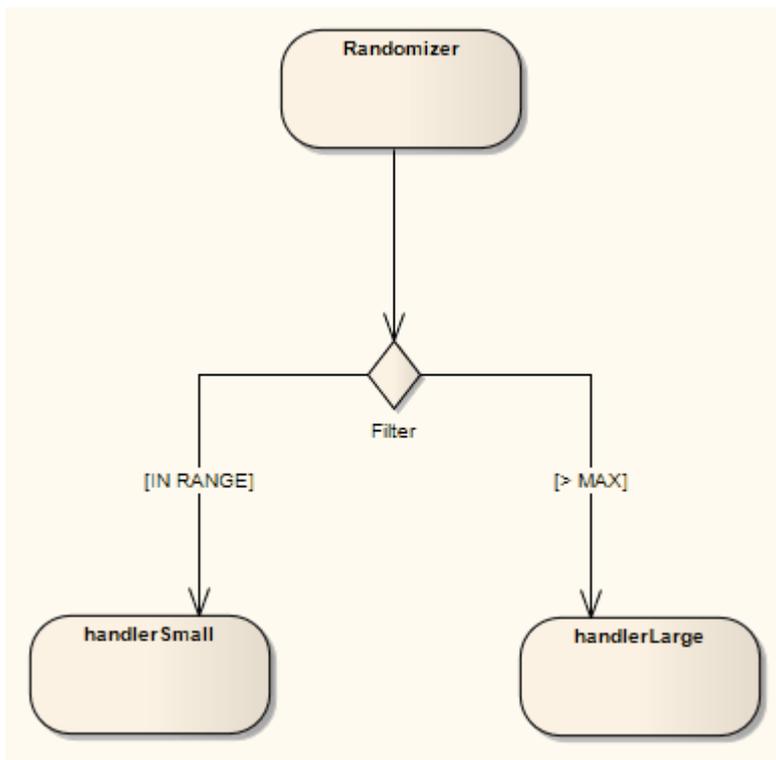
6.4.1.5 Choice



Description

The Choice pseudo-state is used to compose complex transitional paths in, for example, a State Machine diagram, where the outgoing transition path is decided by dynamic, run-time conditions. The run-time conditions are determined by the actions performed by the State Machine on the path leading to the choice.

The following example depicts the Choice element. Upon reaching the Filter pseudo-state, a transition fires to the appropriate state based on the run-time value passed to the Filter. Very similar in form to a Junction pseudo-state, the Choice pseudo-state's distinction is in deciding transition paths at run-time.



Toolbox Icon



Learn More:

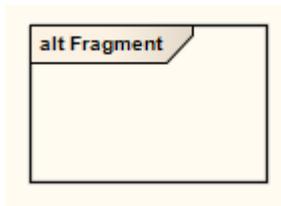
- [State Machine Diagram](#)^[815]
- [State Machine](#)^[917]
- [Pseudo-states](#)^[820]
- [Junction](#)^[908]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

...choice vertices which, when reached, result in the dynamic evaluation of the guards of the triggers of its outgoing transitions. This realizes a dynamic conditional branch. It enables splitting of transitions into multiple outgoing paths such that the decision on which path to take may be a function of the results of prior actions performed in the same run-to-completion step. If more than one of the guards evaluates to true, an arbitrary one is selected. If none of the guards evaluates to true, then the model is considered ill-formed. (To avoid this, it is recommended to define one outgoing transition with the predefined "else" guard for every choice vertex.) Choice vertices should be distinguished from static branch points that are based on junction points.

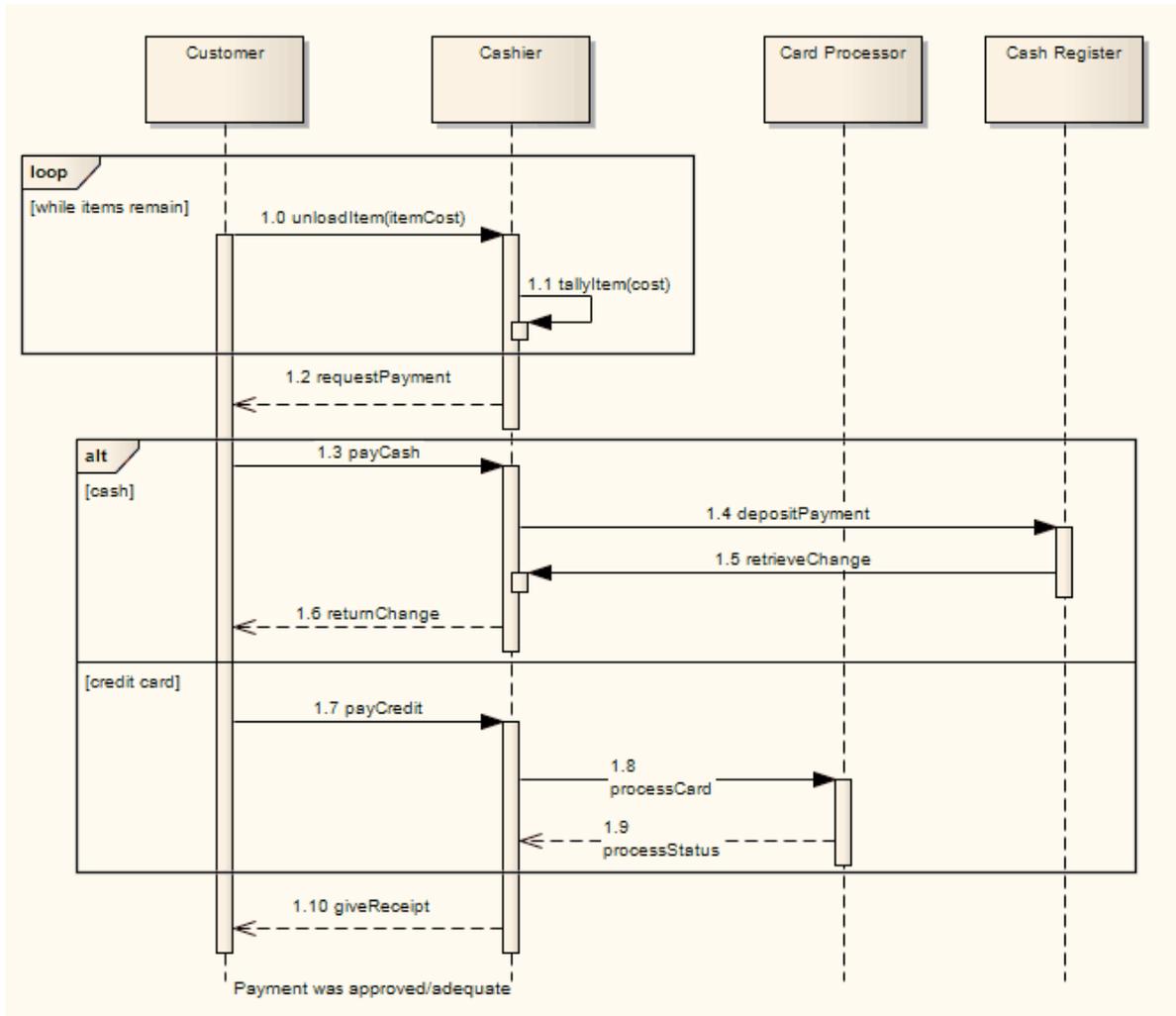
6.4.1.6 Combined Fragment



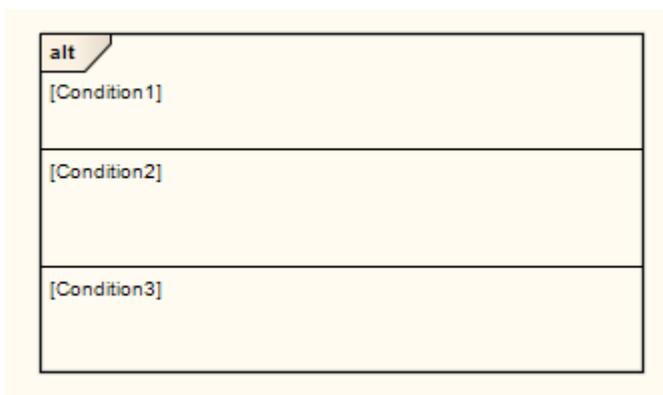
Description

A Combined Fragment reflects a piece or pieces of interaction (called interaction operands) controlled by an interaction operator, whose corresponding boolean conditions are known as interaction constraints. It displays as a transparent window, divided by horizontal dashed lines for each operand.

The following diagram illustrates the use of Combined Fragments, with a Sequence diagram modeling a simplified purchasing process. A loop fragment is created to iterate through an unknown number of items for purchase, after which the cashier requests payment. At this point, two payment options are considered and an alternative fragment is created, divided to show the two operands: cash and credit card. After the fragment completes its trace, the cashier gives a receipt to the customer, under the fulfilled condition that payment requirements were met.



The order of interaction fragment conditions can be changed directly on the diagram. Select an interaction fragment with more than one condition defined. Up and down arrows appear on the right hand side of the each condition. Just click on the arrow to change the order.



Notes:

- In order to select an interaction fragment, you must click near the inside edge or drag a selection rectangle around the fragment; this prevents accidental selection when moving connectors inside the interaction fragment

- Press and hold (**Alt**) to move a combined fragment independently of its contents

Toolbox Icon:



Learn More:

- [Sequence Diagram](#) ^[849]
- [Create a Combined Fragment](#) ^[882]
- [Interaction Operator](#) ^[882]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 467*) states:

A combined fragment defines an expression of interaction fragments. A combined fragment is defined by an interaction operator and corresponding interaction operands. Through the use of CombinedFragments the user will be able to describe a number of traces in a compact and concise manner.

6.4.1.6.1 Create a Combined Fragment

How to:

To create a Combined Fragment in Enterprise Architect, follow the steps below:

Step	Action	See Also
1	Drag the <i>Fragment</i> element from the Interaction Elements page of the Toolbox	Combined Fragment ^[880]
2	In the Type field, click on the drop-down arrow and select one of the various types of interaction operator	Interaction Operators ^[882]
3	In the Condition field, specify a condition or interaction constraint for each operand	
4	A rectangular frame displays, partitioned by dashed lines into segments for each operand	
5	Adjust the frame to encompass the required event occurrences for each operand	

6.4.1.6.2 Interaction Operators

When creating Combined Fragments, you must apply an appropriate interaction operator to characterize the fragment. The following table provides guidance on the various operators, and their corresponding descriptions.

Interaction Operator	Action
alt	Divide up interaction fragments based on Boolean conditions.

Interaction Operator	Action
opt	Enclose an optional fragment of interaction.
par	Indicate that operands operate in parallel.
loop	Indicate that the operand repeats a number of times, as specified by interaction constraints.
critical	Indicate a sequence that cannot be interrupted by other processing.
neg	Assert that a fragment is invalid, and implies that all other interaction is valid.
assert	Specify the only valid fragment to occur. Often enclosed within a <i>consider</i> or <i>ignore</i> operand.
strict	Indicate that the behaviors of the operands must be processed in strict sequence.
seq	Indicate that the Combined Fragment is weakly sequenced. This means that the ordering within operands is maintained, but the ordering between operands is undefined, so long as an event occurrence of the first operand precedes that of the second operand, if the event occurrences are on the same lifeline.
ignore	Indicate which messages should be ignored during execution, or can appear anywhere in the execution trace.
consider	Specify which messages should be considered in the trace. This is often used to specify the resulting event occurrences with the use of an assert operator.
ref	Provide a reference to another diagram. The ref fragment is not created using the method described in the <i>Create a Combined Fragment</i> topic. To create a ref fragment, simply drag an existing diagram from the Project Browser onto the current diagram.

See Also

- [Combined Fragments](#)^[880]
- [Create a Combined Fragment](#)^[882]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 468-471*) states:

The semantics of a CombinedFragment is dependent upon the interactionOperator as explained below.

Alternatives

*The interactionOperator **alt** designates that the CombinedFragment represents a choice of behavior. At most one of the operands will be chosen. The chosen operand must have an explicit or implicit guard expression that evaluates to true at this point in the interaction. An implicit true guard is implied if the operand has no guard.*

The set of traces that defines a choice is the union of the (guarded) traces of the operands.

*An operand guarded by **else** designates a guard that is the negation of the disjunction of all other guards in the enclosing CombinedFragment.*

If none of the operands has a guard that evaluates to true, none of the operands are executed and the remainder of the enclosing InteractionFragment is executed.

Option

The interactionOperator **opt** designates that the CombinedFragment represents a choice of behavior where either the (sole) operand happens or nothing happens. An option is semantically equivalent to an alternative CombinedFragment where there is one operand with non-empty content and the second operand is empty.

Break

The interactionOperator **break** designates that the CombinedFragment represents a breaking scenario in the sense that the operand is a scenario that is performed instead of the remainder of the enclosing InteractionFragment. A **break** operator with a guard is chosen when the guard is true and the rest of the enclosing Interaction Fragment is ignored. When the guard of the **break** operand is false, the **break** operand is ignored and the rest of the enclosing InteractionFragment is chosen. The choice between a **break** operand without a guard and the rest of the enclosing InteractionFragment is done non-deterministically.

A CombinedFragment with interactionOperator **break** should cover all Lifelines of the enclosing InteractionFragment.

Parallel

The interactionOperator **par** designates that the CombinedFragment represents a parallel merge between the behaviors of the operands. The OccurrenceSpecifications of the different operands can be interleaved in any way as long as the ordering imposed by each operand as such is preserved.

A parallel merge defines a set of traces that describes all the ways that OccurrenceSpecifications of the operands may be interleaved without obstructing the order of the OccurrenceSpecifications within the operand.

Weak Sequencing

The interactionOperator **seq** designates that the CombinedFragment represents a weak sequencing between the behaviors of the operands.

Weak sequencing is defined by the set of traces with these properties:

1. The ordering of OccurrenceSpecifications within each of the operands is maintained in the result.
2. OccurrenceSpecifications on different lifelines from different operands may come in any order.
3. OccurrenceSpecifications on the same lifeline from different operands are ordered such that an OccurrenceSpecification of the first operand comes before that of the second operand.

Thus weak sequencing reduces to a parallel merge when the operands are on disjunct sets of participants. Weak sequencing reduces to strict sequencing when the operands work on only one participant.

Strict Sequencing

The interactionOperator **strict** designates that the CombinedFragment represents a strict sequencing between the behaviors of the operands. The semantics of strict sequencing defines a strict ordering of the operands on the first level within the CombinedIFragment with interactionOperator **strict**. Therefore OccurrenceSpecifications within contained CombinedFragment will not directly be compared with other OccurrenceSpecifications of the enclosing CombinedFragment.

Negative

The interactionOperator **neg** designates that the CombinedFragment represents traces that are defined to be invalid.

The set of traces that defined a CombinedFragment with interactionOperator negative is equal to the set of traces given by its (sole) operand, only that this set is a set of invalid rather than valid traces. All InteractionFragments that are different from Negative are considered positive meaning that they describe traces that are valid and should be possible.

Critical Region

The interactionOperator **critical** designates that the CombinedFragment represents a critical region. A critical region means that the traces of the region cannot be interleaved by other OccurrenceSpecifications (on those Lifelines covered by the region). This means that the region is treated atomically by the enclosing

fragment when determining the set of valid traces. Even though enclosing CombinedFragments may imply that some OccurrenceSpecifications may interleave into the region, such as with **par**-operator, this is prevented by defining a region.

Thus the set of traces of enclosing constructs are restricted by critical regions.

Ignore / Consider

(p. 473) The interactionOperator **ignore** designates that there are some message types that are not shown within this combined fragment. These message types can be considered insignificant and are implicitly ignored if they appear in a corresponding execution. Alternatively one can understand **ignore** to mean that the messages that are ignored can appear anywhere in the traces.

Conversely the interactionOperator **consider** designates which messages should be considered within this CombinedFragment. This is equivalent to defining every other message to be ignored.

Assertion

The interactionOperator **assert** designates that the CombinedFragment represents an assertion. The sequences of the operand of the assertion are the only valid continuations. All other continuations result in an invalid trace. Assertions are often combined with Ignore or Consider.

Loop

The interactionOperator **loop** designates that the CombinedFragment represents a loop. The **loop** operand will be repeated a number of times.

The Guard may include a lower and an upper number of iterations of the loop as well as a Boolean expression. The semantics is such that a loop will iterate minimum the 'minint' number of times (given by the iteration expression in the guard) and at most the 'maxint' number of times. After the minimum number of iterations have executed, and the boolean expression is false the loop will terminate. The loop construct represent a recursive application of the **seq** operator where the **loop** operand is sequenced after the result of earlier iterations.

The Semantics of Gates

The gates of a CombinedFragment represent the syntactic interface between the CombinedFragment and its surroundings, which means the interface towards other InteractionFragments.

The only purpose of gates is to define the source and the target of messages.

6.4.1.7 Datastore



Description

A Datastore is an element used to define permanently stored data. A token of data that enters into a Datastore is stored permanently, updating tokens for data that already exists. A token of data that comes out of a Datastore is a copy of the original data.

Use Object Flow connectors to connect elements (such as Activities) to Datastores, as values and information are being passed between nodes. Selection and transformation behavior, together composing a sort of query, can be specified as to the nature of data access. For instance, selection behavior determines which objects are affected by the connection to the Datastore. Transformation behavior might then further specify the value of an attribute pertaining to a selected object.

To define the behavior of access to a Datastore, attach a note to the Object Flow connector. To do this, right-click on the Object Flow and select the Attach Note or Constraint context menu option. A dialog indicates

other flows in the Activity diagram, to which you can attach the note (if the behavior applies to multiple flows). To comply with UML 2, preface behavior with the notation «selection» or «transformation».

Toolbox Icon



Learn More:

- [Activity Diagram](#)^[81]
- [Activity](#)^[873]
- [Object Flow](#)^[1003]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 360*) states:

A data store node is a central buffer node for non-transient information... A data store keeps all tokens that enter it, copying them when they are chosen to move downstream. Incoming tokens containing a particular object replace any tokens in the object node containing that object.

6.4.1.8 Decision

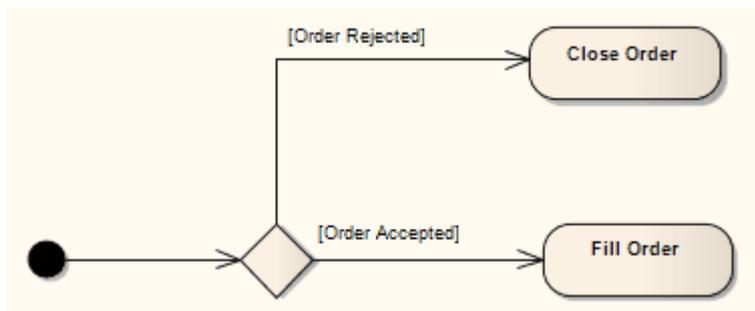


Description

A Decision is an element of an Activity diagram or Interaction Overview diagram that indicates a point of conditional progression: if a condition is true, then processing continues one way; if not, then another.

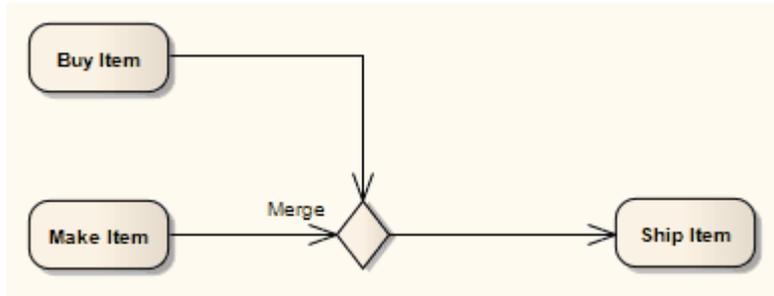
This can also be used as a Merge node in that multiple alternative flows can be merged (but not synchronized) to form one flow. The following examples show both of these modes of using the decision element.

Used as a decision:



See *UML Superstructure Specification, v2.1.1, figure 12.77, p. 363*.

Used as a merge:



See *UML Superstructure Specification, v2.1.1, figure 12.106, p. 388.*

Notes:

- Moving a diagram generally does not affect the location of elements in packages. If you move a diagram out of one package into another, all the elements in the diagram remain in the original package. However, Decision elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram. Therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram.

Toolbox Icon



Learn More:

- [Activity Diagram](#) ^[81]
- [Interaction Overview Diagram](#) ^[86]
- [Merge Node](#) ^[91]

OMG UML Specification

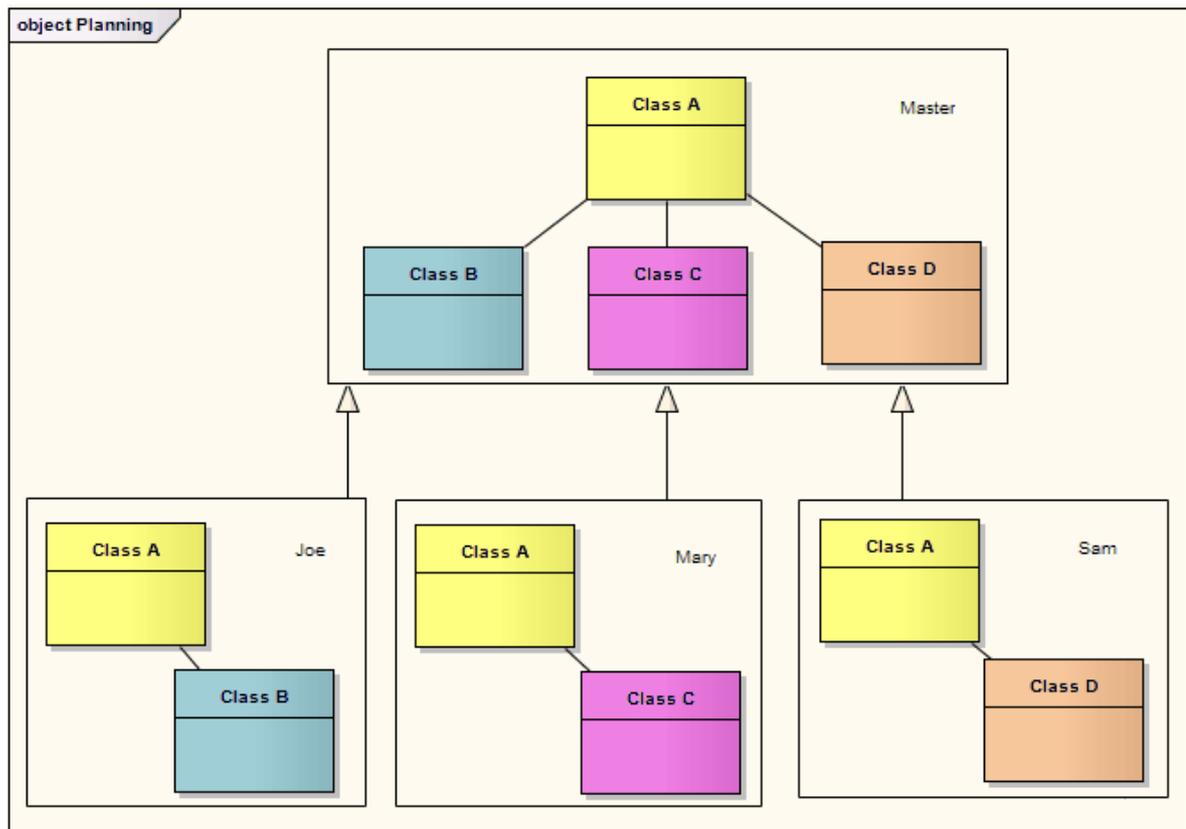
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 361 (Decision symbol)*) states:

A decision node is a control node that chooses between outgoing flows. A decision node has one incoming edge and multiple outgoing activity edges.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 387 (Merge symbol)*) also states:

A merge node is a control node that brings together multiple alternate flows. It is not used to synchronize concurrent flows but to accept one among several alternate flows... A merge node has multiple incoming edges and a single outgoing edge.

6.4.1.9 Diagram Frame



Description

A Diagram Frame element is a rendition of a diagram dropped from the Project Browser into another diagram. It is a type of Combined Fragment with the Interaction Operator ref. However, it can be created on any type of diagram, and is not created in the same way as other Combined Fragments.

When you drop the diagram from the Project Browser onto the open diagram, a dialog shows providing the following options:

- Diagram Frame
- Diagram Reference
- Hyperlink

If you click on the Diagram Frame radio button, a Diagram Frame is inserted into the diagram, containing an image of the dropped diagram.

If you select the Diagram Reference option, an empty frame is inserted with the name of the dropped diagram in the frame label. If you select the Hyperlink radio button, a diagram icon is inserted with no frame, and with the parent package and diagram name next to it.

In all three cases, the object acts as a hyperlink to the real referenced diagram. You can also define properties for the objects, as for other elements, by right-clicking on the object and selecting the element Properties context menu option.

Notes:

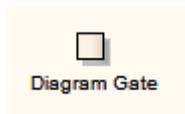
- You can change the size of all three objects, but you cannot reduce a Diagram Frame to less than the size of the enclosed diagram.

- You cannot change the diagram within a Diagram Frame. To edit the diagram, double-click within the frame and edit the original diagram.
- The Diagram Frame element looks identical to but is not the same as a diagram frame border, which you can set automatically on new images of diagrams using the Tools | Options | Diagram option, and selecting the appropriate checkboxes in the Diagram Frames panel. These options set frames on print-outs of diagrams, images of diagrams copied to file, and images of diagrams copied to the clipboard. If you paste the image from the clipboard into another diagram, the image initially looks the same as the Diagram Frame element but it is actually a discreet unit that you manipulate using the Image Manager.

Learn More:

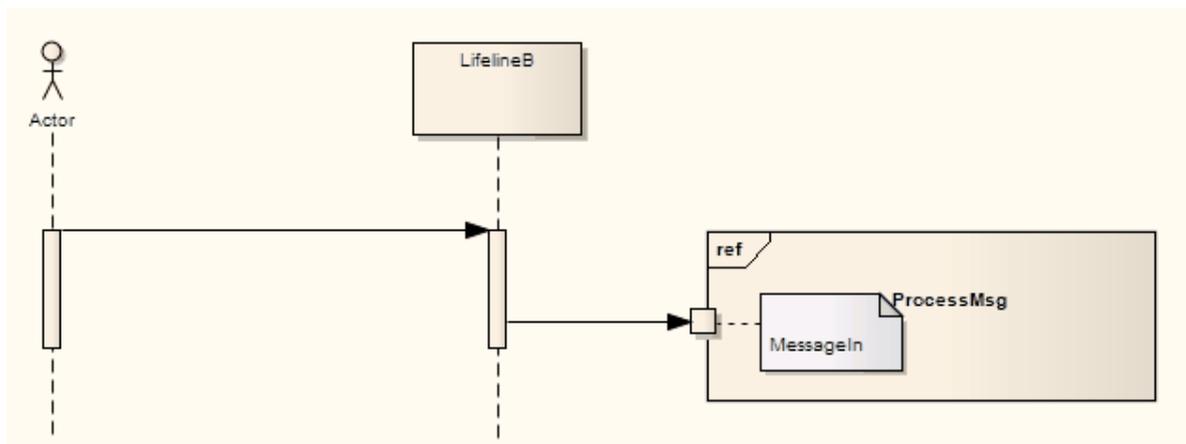
- [Combined Fragment](#)^[880]
- [Interaction Operator: ref](#)^[883]

6.4.1.10 Diagram Gate



Description

A Diagram Gate is a simple graphical way to indicate the point at which messages can be transmitted into and out of interaction fragments. A fragment might be required to receive or deliver a message; internally, an ordered message reflects this requirement, with a gate indicated on the boundary of the fragment's frame. Any external messages 'synching' with this internal message must correspond appropriately. Gates can appear on Interaction diagrams (Sequence, Timing, Communication or Interaction Overview), interaction occurrences and combined fragments (to specify the expression).



Toolbox Icon



Learn More:

- [Sequence Diagram](#) ^[849]
- [Timing Diagram](#) ^[830]
- [Communication Diagram](#) ^[859]
- [Interaction Overview Diagram](#) ^[861]
- [Interaction Occurrence](#) ^[906]
- [Combined Fragment](#) ^[880]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 480*) states:

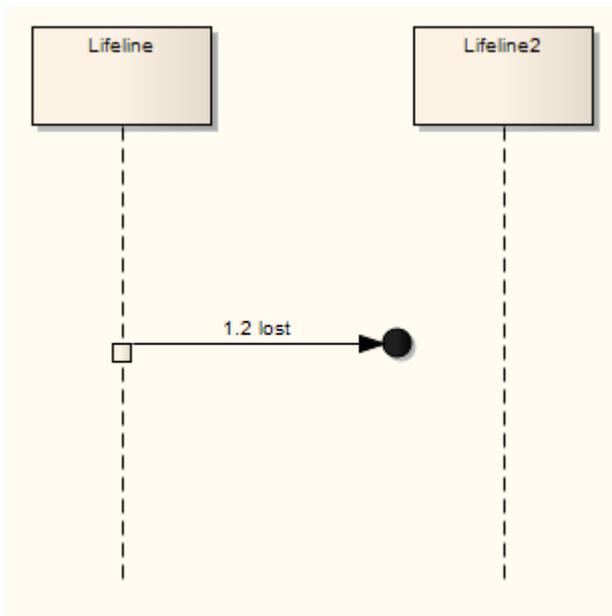
A Gate is a connection point for relating a Message outside an InteractionFragment with a Message inside the InteractionFragment ... Gates are connected through Messages. A Gate is actually a representative of an OccurrenceSpecification that is not in the same scope as the Gate. Gates play different roles: we have formal gates on Interactions, actual gates on InteractionUses, expression gates on CombinedFragments.

6.4.1.11 Endpoint**Description**

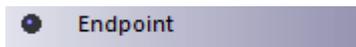
An Endpoint is used in Interaction diagrams (Sequence, Timing, Communication or Interaction Overview) to reflect a lost or found message in sequence. To model this, drag an Endpoint element onto the workspace.

With Sequence diagrams, drag a message from the appropriate lifeline to the Endpoint. With Timing diagrams, the message connecting the lifeline to the Endpoint requires some timing specifications to draw the connection.

The following example depicts a lost message in a Sequence diagram.



Toolbox Icon



Learn More:

- [Sequence Diagram](#) ^[849]
- [Timing Diagram](#) ^[830]
- [Communication Diagram](#) ^[859]
- [Interaction Overview Diagram](#) ^[861]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 492*) states:

A lost message is a message where the sending event occurrence is known, but there is no receiving event occurrence. We interpret this to be because the message never reached its destination.

A found message is a message where the receiving event occurrence is known, but there is no (known) sending event occurrence. We interpret this to be because the origin of the message is outside the scope of the description. This may, for example, be noise or other activity that we do not want to describe in detail.

6.4.1.12 Entry Point



Description

Entry Point pseudo-states are used to define the beginning of a State Machine. An Entry Point exists for each

region, directing the initial concurrent state configuration.

Toolbox Icon



Learn More:

- [State Machine Diagram](#) ^[815]
- [Pseudo-states](#) ^[820]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 471*) states:

An entry point pseudostate is an entry point of a state machine or composite state. In each region of the state machine or composite state it has a single transition to a vertex within the same region.

6.4.1.13 Exception



Description

The Exception Handler element defines the group of operations to carry out when an exception occurs. In an Activity diagram, the protected element can contain a set of operations and is connected to the exception handler via an Interrupt Flow connector. Any defined error contained within an element's parts can trigger the flow to move to an exception.

Toolbox Icon



Learn More:

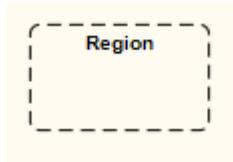
- [Activity Diagram](#) ^[811]
- [Interrupt Flow](#) ^[988]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 364*) states:

An exception handler is an element that specifies a body to execute in case the specified exception occurs during the execution of the protected node.

6.4.1.14 Expansion Region



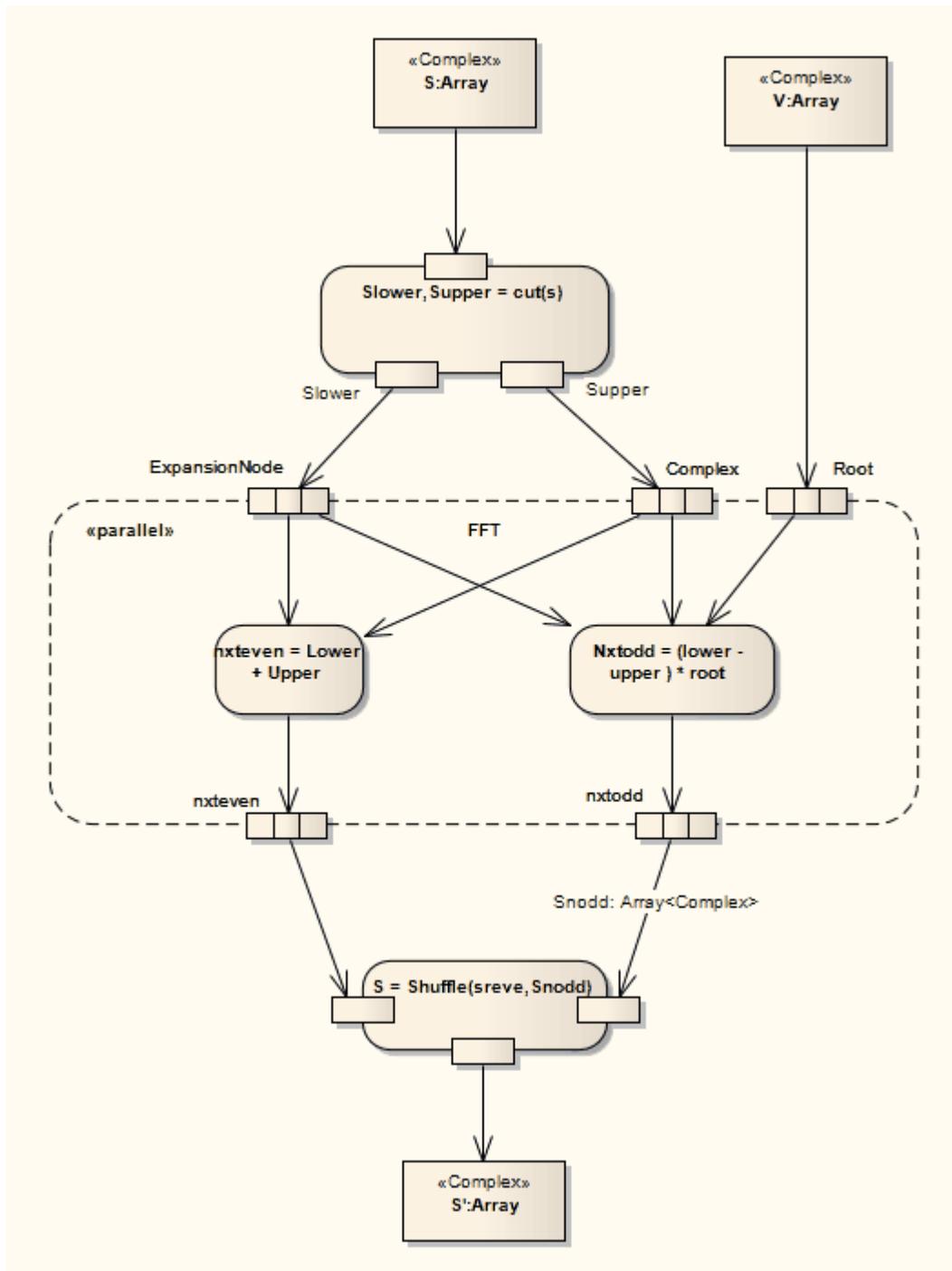
Description

You create an Expansion Region as one variant of a Region (the other is an Interruptible Activity Region).

On an Activity diagram, an Expansion Region surrounds a process to be imposed multiple times on the incoming data, once for every element in the input collection. If there are multiple inputs, the collection sizes must match, and the elements within each collection must be of the same type. Similarly, any outputs must be in the form of a collection matching the size of the inputs.

The concurrency of the Expansion Region's multiple executions can be specified as type parallel, iterative, or stream. Parallel reflects that the elements in the incoming collections can be processed at the same time or overlapping, whereas an iterative concurrency type specifies that execution must occur sequentially. A stream-type Expansion Region indicates that the input and output come in and exit as streams, and that the Expansion Region's process must have some method to support streams.

To modify the mode of an Expansion Region, right-click on it and select the **Properties** context menu option, then select the Advanced page.



See *UML Superstructure Specification, v2.1.1, figure 12.87, p. 372.*

Toolbox Icon



Learn More:

- [Activity Diagram](#)^[817]
- [Region](#)^[916]
- [Interruptible Activity Region](#)^[907]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 367*) states:

An expansion region is a structured activity region that executes multiple times corresponding to elements of an input collection.

6.4.1.15 Exit Point



Description

Exit Points are used in Submachine states and State Machines to denote the point where the machine is exited and the transition sourcing this exit point, for Submachines, is triggered. Exit points are a type of pseudo-state used in the State Machine diagram.

Toolbox Icon



Learn More:

- [State Machine Diagram](#)^[815]
- [State Element](#)^[917]
- [Submachine State](#)^[925]
- [Pseudo-state](#)^[820]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1 p. 538*) states:

An exit point pseudostate is an exit point of a state machine or composite state. Entering an exit point within any region of the composite state or state machine referenced by a submachine state implies the exit of this composite state or submachine state and the triggering of the transition that has this exit point as source in the state machine enclosing the submachine or composite state.

6.4.1.16 Final

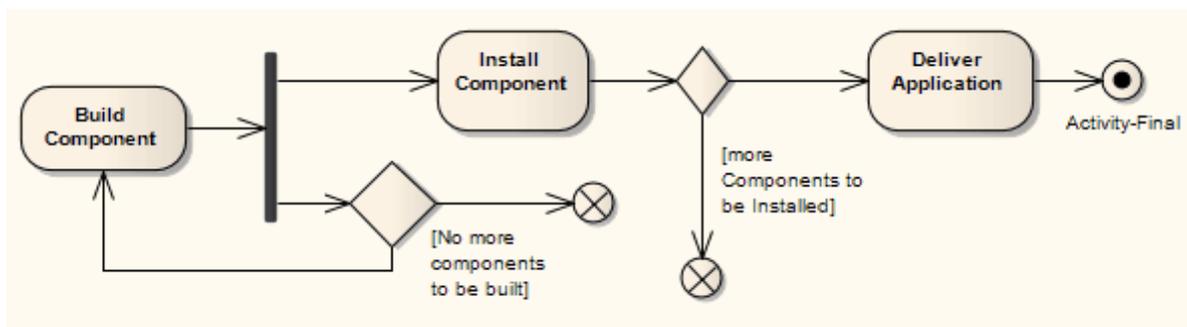


Description

There are two nodes used to define a Final state in an Activity, both defined in UML 2.3 as of type Final Node. The Activity Final element, shown above, indicates the completion of an Activity; upon reaching the Final, all execution in the Activity diagram is aborted. The other type of final node, Flow Final, depicts an exit from the system that has no effect on other executing flows in the Activity.

The following example illustrates the development of an application. The process comes to a Flow Final node when there are no more components to be built; note that the Fork element indicates a concurrent process with the building of new components and installation of completed components. The Flow Final terminates only the sub-process building components. Similarly, only those tokens entering the decision branch for the installation of further components terminate with the connecting Flow Final (that is, stop installing this component, but keep on installing other components). It is only after the Deliver Application activity is completed, after the control flow reaches the Final node, that all flows stop.

The node that initiates a flow is the Initial node.



See *UML Superstructure Specification, v2.1.1, figure 12.91, p. 374.*

Notes:

- Moving a diagram generally does not affect the location of elements in packages; if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Final elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Toolbox Icon



Learn More:

- [Activity Diagram](#)^[81]
- [Activity](#)^[87]
- [Flow Final](#)^[89]
- [Fork](#)^[90]
- [Initial](#)^[90]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 332*) states:

An activity may have more than one activity final node. The first one reached stops all flows in the activity.

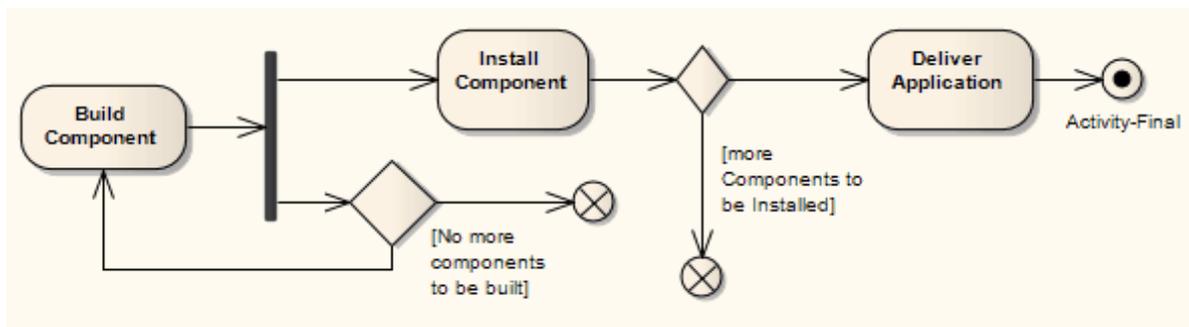
6.4.1.17 Flow Final



Description

There are two nodes used to define a final state in an Activity, both defined in UML 2.3 as of type Final Node. The Flow Final element depicts an exit from the system, as opposed to the Activity Final, which represents the completion of the Activity. Only the flow entering the Flow Final node exits the Activity; other flows continue undisturbed.

The following example Activity Diagram illustrates the development of an application. The process comes to a Flow Final node when there are no more components to be built; note that the Fork element indicates a concurrent process with the building of new components and installation of completed components. The Flow Final terminates only the sub-process building components. Similarly, only those tokens entering the decision branch for the installation of further components terminate with the connecting Flow Final (that is, stop installing this component, but keep on installing other components). It is only after the Deliver Application activity is completed, after the control flow reaches the Final node, that all flows stop.



See *UML Superstructure Specification, v2.1.1, figure 12.91, p. 374*.

Notes:

- Moving a diagram generally does not affect the location of elements in packages: if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Flow Final elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Toolbox Icon

 Flow Final
Learn More:

- [Activity Diagram](#) ^[817]
- [Activity Final](#) ^[895]
- [Fork](#) ^[900]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 375*) states:

A flow final destroys all tokens that arrive at it. It has no effect on other flows in the activity.

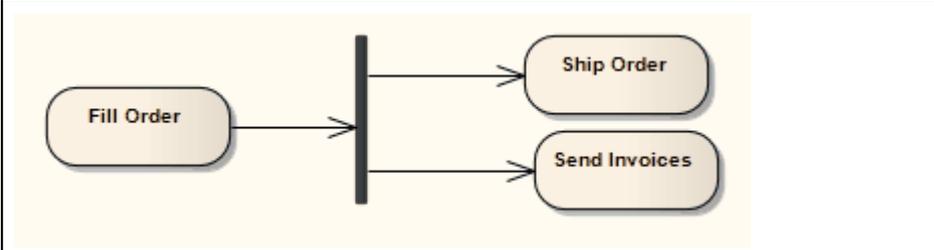
6.4.1.18 Fork/Join**Description**

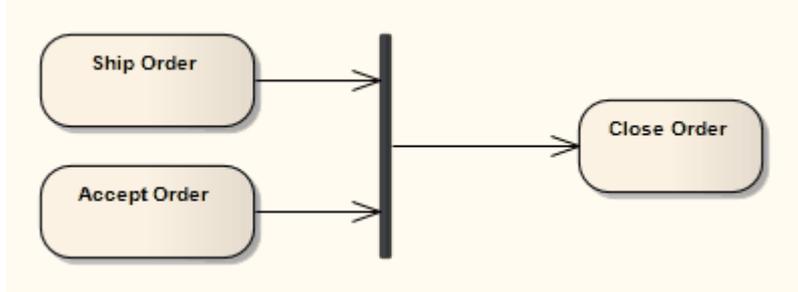
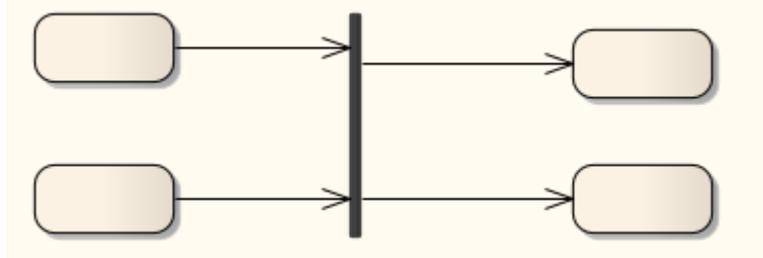
The *Fork/Join* elements have different modes of use, as follows:

- To fork or split the flow into a number of concurrent flows.
- To join the flow of a number of concurrent flows.
- To both join and fork a number of incoming flows to a number of outgoing flows.

These elements are used in both Activity and State Machine diagrams. With respect to State Machine diagrams, Forks and Joins are used as pseudo-states. Other pseudo-states include history states, entry points and exit points. Forks are used to split an incoming transition into concurrent multiple transitions leading to different target states. Joins are used to merge concurrent multiple transitions into a single transition leading to a single target. They are semantic inverses. To learn more about these individual elements see their specific topics.

Example Diagrams:

Description	Diagram
Fork or split the flow into a number of concurrent flows	 <p>The diagram illustrates a Fork element (a vertical bar) used to split a flow. An activity state 'Fill Order' has an outgoing arrow that points to a vertical bar. From the right side of this bar, two arrows branch out to two separate activity states: 'Ship Order' and 'Send Invoices'.</p>

Description	Diagram
Join the flow of a number of concurrent flows	
Join and Fork a number of incoming flows to a number of outgoing flows	

Toolbox Icon



See Also:

- [Fork](#) ^[900]
- [Join](#) ^[901]

Learn More:

- [Activity Diagram](#) ^[811]
- [State Machine Diagram](#) ^[815]
- [Pseudo-state](#) ^[820]

OMG UML Specification

Fork

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 376*) states:

A fork node is a control node that splits a flow into multiple concurrent flows... A fork node has one incoming edge and multiple outgoing edges.

Join

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 381-382*) states:

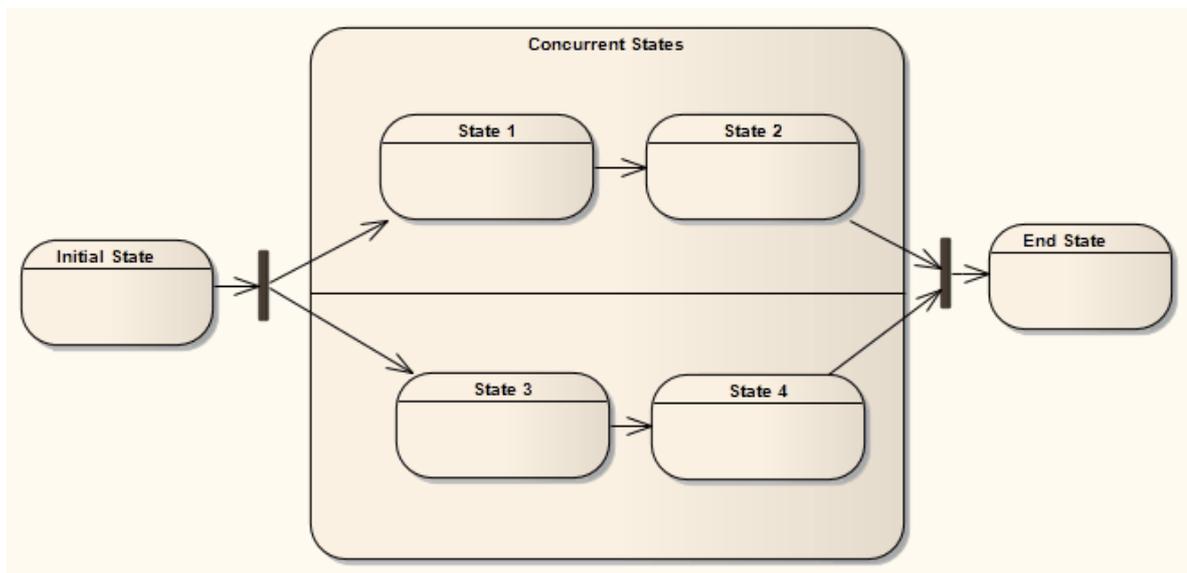
A join node is a control node that synchronizes multiple flows... A join node has multiple incoming edges and one outgoing edge.

6.4.1.18.1 Fork



Description:

The Fork element is used in both Activity and State Machine diagrams. With respect to State Machine diagrams, a Fork pseudo-state signifies that its incoming transition comes from a single state, and it has multiple outgoing transitions. These transitions must occur concurrently, requiring the use of concurrent regions, as depicted below in the Composite State. Unlike Choice or Junction pseudo-states, Forks must not have triggers or guards. The following diagram demonstrates a Fork pseudo-state dividing into two concurrent regions, which then return to the End State via the Join pseudo-state.



Learn More:

- [Activity Diagram](#) ^[817]
- [State Machine Diagram](#) ^[815]
- [Pseudo-state](#) ^[820]
- [States](#) ^[917]
- [Choice](#) ^[879]
- [Junction](#) ^[908]
- [Join](#) ^[907]
- [Region](#) ^[916]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

Fork vertices serve to split an incoming transition into two or more transitions terminating on orthogonal target vertices (i.e. vertices in different regions of a composite state). The segments outgoing from a fork

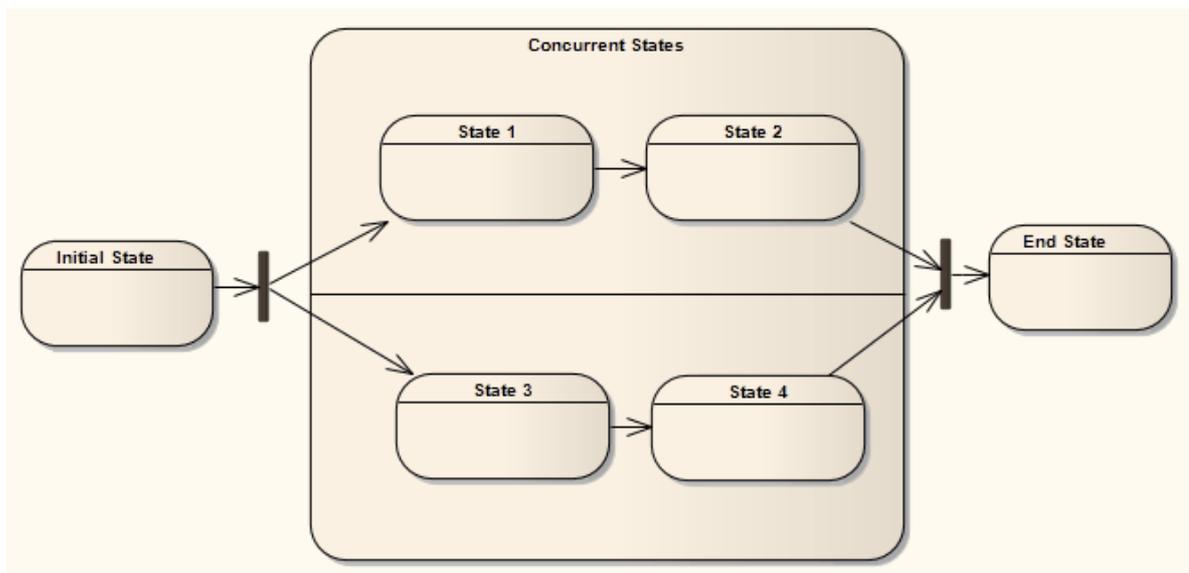
vertex must not have guards or triggers.

6.4.1.18.2 Join



Description

The Join element is used by Activity and State Machine diagrams. The above example illustrates a Join transition between Activities. With respect to State Machine diagrams, a Join pseudo-state indicates multiple States concurrently transitioning into the Join and onto a single State. Unlike Choice or Junction pseudo-states, Joins must not have triggers or guards. The following diagram demonstrates a Fork pseudo-state dividing into two concurrent Regions, which then return to the End State via the Join.



Learn More:

- [Activity Diagram](#) ^[817]
- [State Machine Diagram](#) ^[815]
- [Pseudo-state](#) ^[820]
- [States](#) ^[917]
- [Choice](#) ^[879]
- [Junction](#) ^[908]
- [Fork](#) ^[900]
- [Region](#) ^[916]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

Join vertices serve to merge several transitions emanating from source vertices in different orthogonal

regions. The transitions entering a join vertex cannot have guards or triggers.

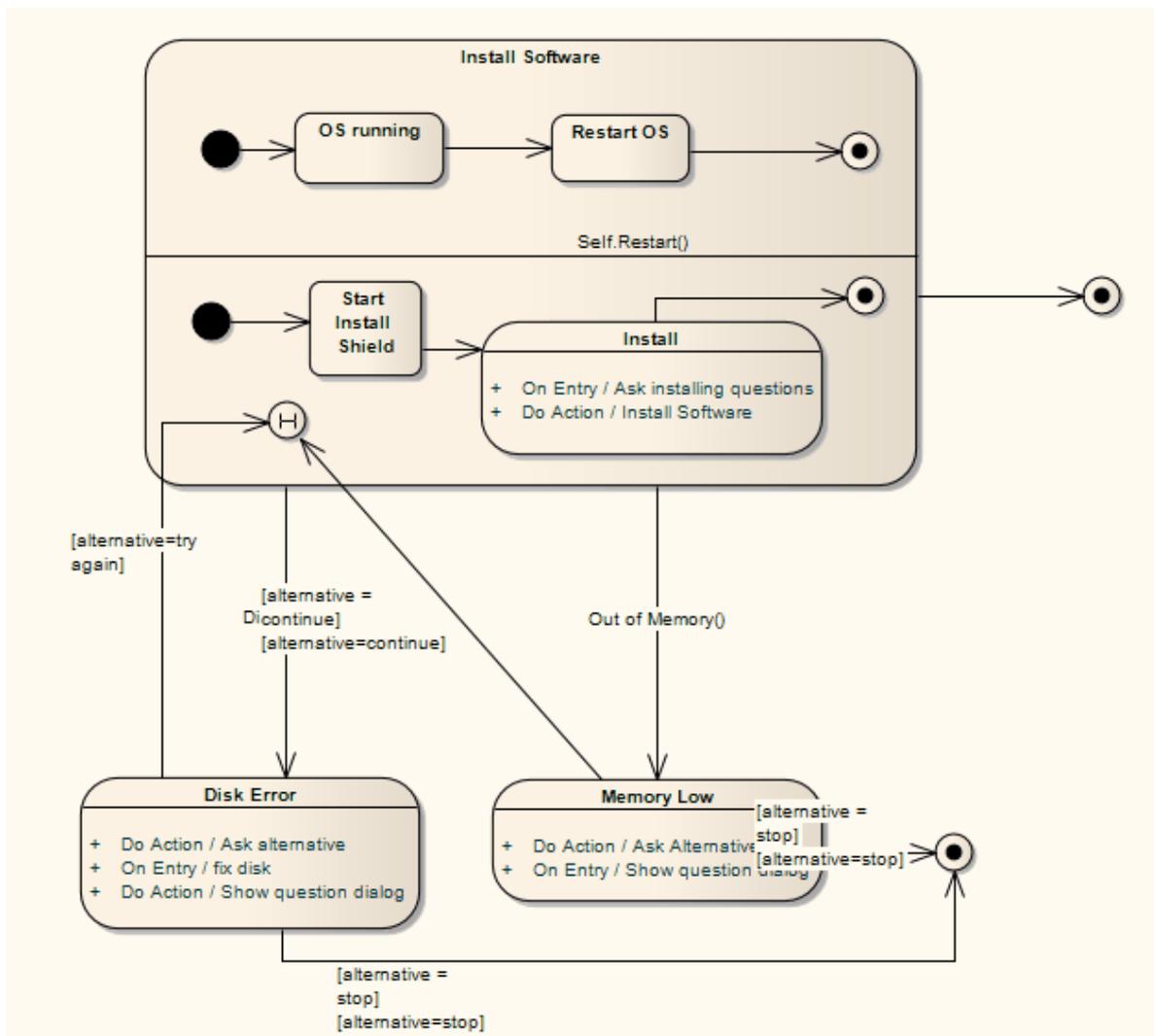
6.4.1.19 History



Description

There are two types of History pseudo-state defined in UML: shallow and deep history. A shallow History sub-state is used to represent the most recently active sub-state of a Composite State; this pseudo-state does not recurse into this sub-state's active configuration, should one exist. A single connector can be used to depict the default shallow History state, in case the Composite State has never been entered.

A deep History sub-state, in contrast, reflects the most recent active configuration of the Composite State. This includes active sub-states of all regions, and recurses into those sub-states' active sub-states, should they exist. At most one deep history and one shallow history can dwell within a composite state. You can reassign a shallow History substate as a deep History substate using the Advanced element context menu.



Toolbox Icon



Learn More:

- [Pseudo-state](#)^[820]
- [Composite State](#)^[919]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 537*) states:

... deepHistory represents the most recent active configuration of the composite state that directly contains this pseudostate (e.g., the state configuration that was active when the composite state was last exited). A composite state can have at most one deep history vertex. At most one transition may originate from the history connector to the default deep history state. This transition is taken in case the composite state had never been active before. Entry actions of states entered on the path to the state represented by a deep history are performed.

... shallowHistory represents the most recent active substate of its containing state (but not the substates of that sub state). A composite state can have at most one shallow history vertex. A transition coming into the shallow history vertex is equivalent to a transition coming into the most recent active substate of a state. At most one transition may originate from the history connector to the default shallow history state. This transition is taken in case the composite state had never been active before. Entry actions of states entered on the path to the state represented by a shallow history are performed.

6.4.1.20 Initial



Description

The Initial element is used by Activity and State Machine diagrams. In Activity diagrams, it defines the start of a flow when an Activity is invoked. With State Machines, the Initial element is a pseudo-state used to denote the default state of a Composite State; there can be one Initial vertex in each Region of the Composite State.

This simple example shows the start of a flow to receive an order.



See *UML Superstructure Specification, v2.1.1, Figure 12.97, p. 378*.

The activity flow is completed by a Final or Flow Final node.

Notes:

- Moving a diagram generally does not affect the location of elements in packages; if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Initial elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

Toolbox Icon**Learn More:**

- [Activity Diagram](#) ^[817]
- [State Machine Diagram](#) ^[815]
- [Activity](#) ^[873]
- [Pseudo-state](#) ^[820]
- [Composite State](#) ^[919]
- [Region](#) ^[916]
- [Final](#) ^[895]
- [Flow Final](#) ^[897]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 537*) states:

An initial pseudostate represents a default vertex that is the source for a single transition to the default state of a composite state. There can be at most one initial vertex in a region. The outgoing transition from the initial vertex may have a behavior, but not a trigger or guard.

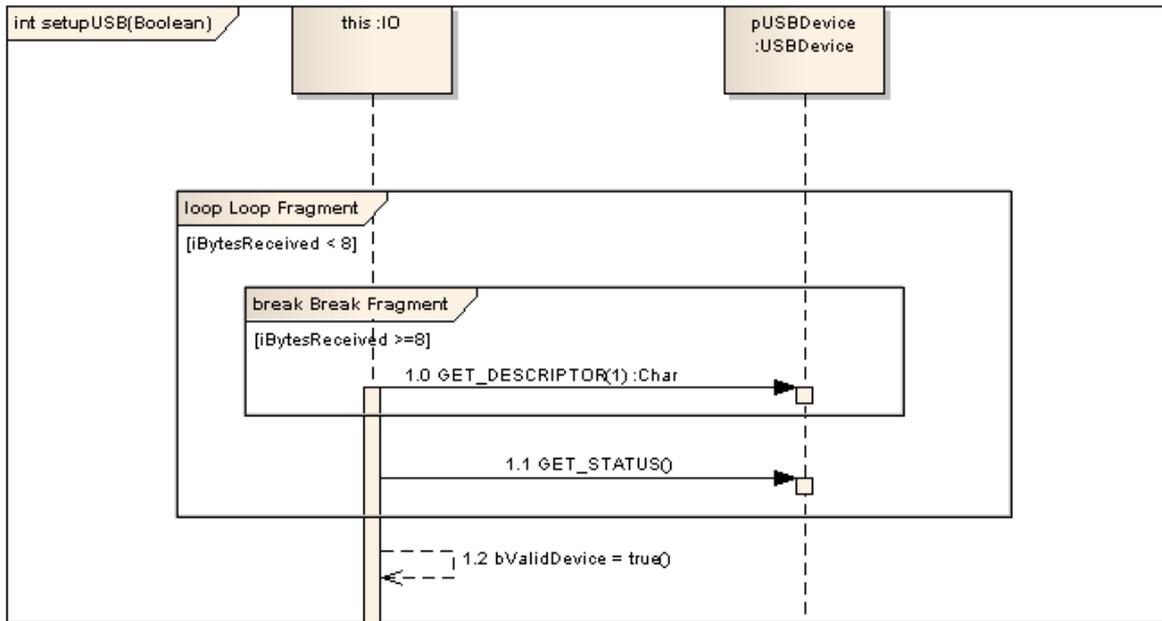
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 378*) also states:

An initial node is a control node at which flow starts when the activity is invoked.

6.4.1.21 Interaction**Description**

An *Interaction* element is used to describe a system, representing its interactions at varying levels of detail, for review not only by design professionals but also by end users and stakeholders. An Interaction element can contain the following types of diagram:

- Sequence
- Interaction Overview
- Communication
- Timing.

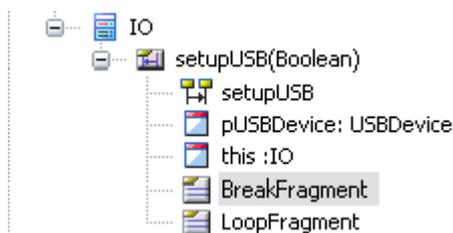


An Interaction element in Enterprise Architect is treated as a behavior of the classifier it is encapsulated within. It can have parameters and return types, which are modeled using the **Behavior** tab of the Interaction element's **Properties** dialog. The element is interpreted as a method of the containing Class in the generated code (see the *Generate Code From Behavioral Model* topic).

An Interaction element can also be set as the classifier for an *Interaction Occurrence in a Sequence diagram*, or for a *Call Behavior Action in an Activity diagram*. Establishing such an association (between a behavior and a behavior call) facilitates adding arguments that can be individually mapped to the associated behavior's parameters.

Notes:

- The behavioral code generation engine expects the Sequence diagram and all its associated messages and interaction fragments to be encapsulated within an Interaction element (such as *setupUSB* in the example below).



(The IO Class shown above is available in the EAExample model, under **Systems Engineering Model | Implementation Model | Software**.)

Learn More:

- [Sequence Diagram](#) ^[849]
- [Interaction Overview Diagram](#) ^[861]
- [Communication Diagram](#) ^[859]
- [Timing Diagram](#) ^[830]
- [Generate Code From Behavioral Model](#) ^[1496]
- [Interaction Occurrence](#) ^[908]

- [Call Behavior Action](#) ^[866]
- [Associate with Different Behavior](#) ^[716]
- [Arguments](#) ^[716]

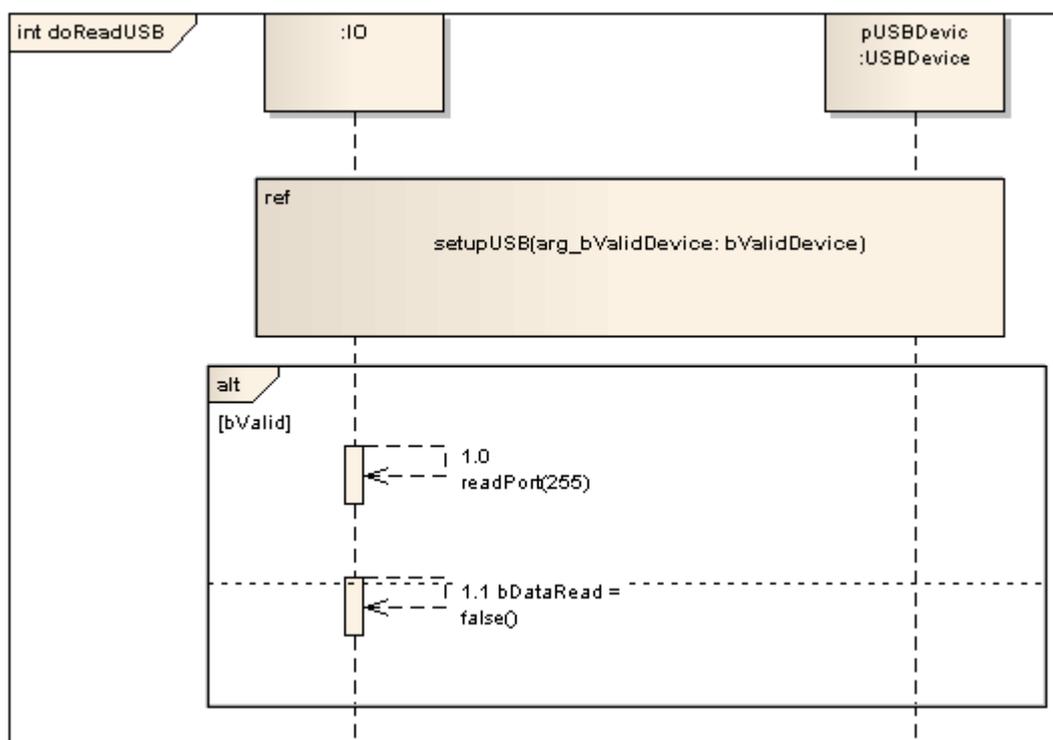
6.4.1.22 Interaction Occurrence



Description

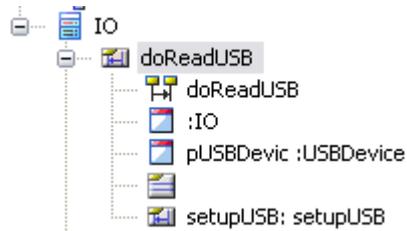
An *Interaction Occurrence* (or *InteractionUse*) is a reference to an existing *Interaction* (Sequence) diagram. Interaction Occurrences are visually represented by a frame, with *ref* in the frame's title space. The diagram name is indicated in the frame contents. To create an Interaction Occurrence, simply open a Sequence diagram (preferably contained within an Interaction element) and drag another Sequence diagram (also preferably contained within an Interaction element) into its workspace. A dialog displays, providing configuration options. The resulting Interaction Occurrence acts as an invocation of the original Interaction. You use the Call tab of the element Properties dialog to set up the actual arguments of the Interaction and also to change to a different associated Interaction element.

The following figure illustrates the use of an Interaction Occurrence in another Interaction (Sequence) diagram. You can display the sequence represented by the Interaction Occurrence by double-clicking on the element.



Notes:

- The behavioral code generation engine expects the Sequence diagram and all its associated messages and interaction fragments to be encapsulated within an Interaction element (such as *doReadUSB* in the example below).



Learn More:

- [Sequence Diagram](#) ^[849]
- [Interaction element](#) ^[904]
- [Behavior Calls Invocations](#) ^[715]
- [Arguments](#) ^[715]

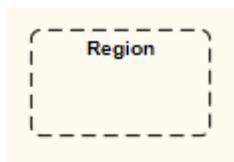
OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 423*) refers to an Interaction Occurrence as an **InteractionUse**, and states:

An InteractionUse refers to an Interaction. The InteractionUse is a shorthand for copying the contents of the referred Interaction where the InteractionUse is. To be accurate the copying must take into account substituting parameters with arguments and connect the formal gates with the actual ones.

It is common to want to share portions of an interaction between several other interactions. An InteractionUse allows multiple interactions to reference an interaction that represents a common portion of their specification.

6.4.1.23 Interruptible Activity Region

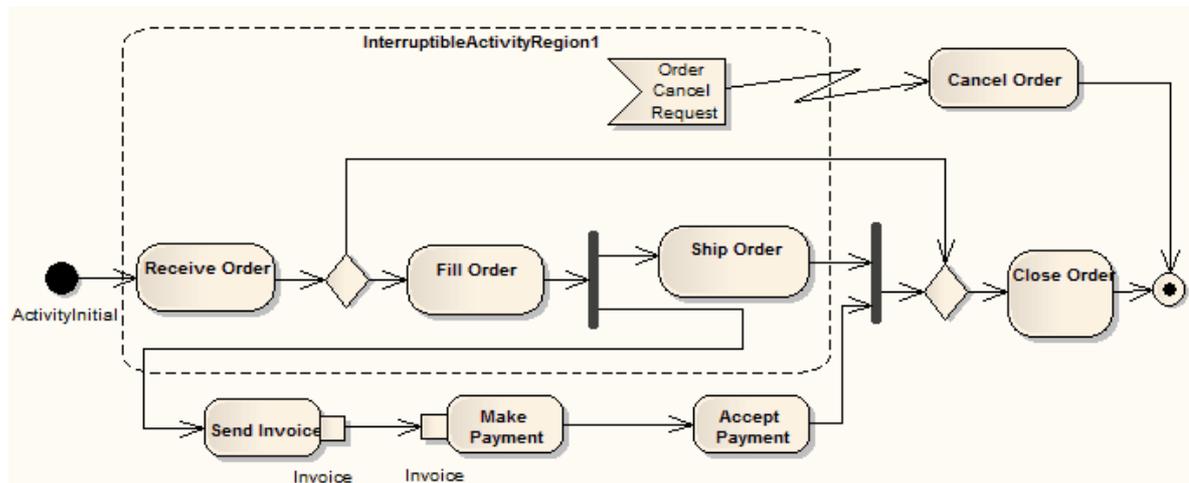


Description

You create an Interruptible Activity Region as one variant of a Region (the other is an Expansion Region).

In an Activity diagram, an Interruptible Activity Region surrounds a group of Activity elements, all affected by certain interrupts in such a way that all tokens passing within the region are terminated should the interruption(s) be raised. Any processing occurring within the bounds of an Interruptible Activity Region is terminated when a flow is instigated across an interrupt flow to an external element.

The example below illustrates that an order cancellation kills any processing of the order at the receipt, filling or shipping stage.



See *UML Superstructure Specification, v2.1.1, figure 12.100, p. 381.*

Toolbox Icon



Learn More:

- [Activity diagram](#) ^[81]
- [Activity](#) ^[873]
- [Region](#) ^[916]
- [Expansion Region](#) ^[893]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 380*) states:

An interruptible region contains activity nodes. When a token leaves an interruptible region via edges designated by the region as interrupting edges, all tokens and behaviors in the region are terminated.

6.4.1.24 Junction

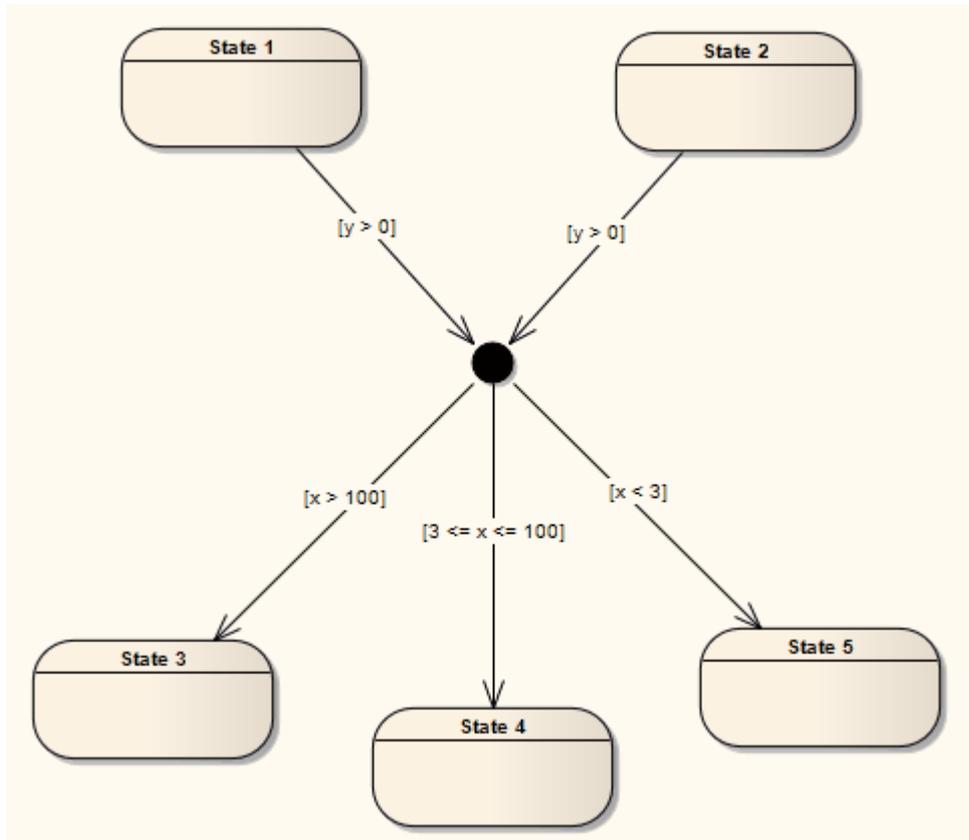


Description

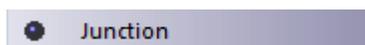
Junction pseudo-states are used to design complex transitional paths in State Machine diagrams. A Junction can be used to combine or merge multiple paths into a shared transition path. Alternatively, a Junction can split an incoming path into multiple paths, similar to a Fork pseudostate. Unlike Forks or Joins, Junctions can apply guards to each incoming or outgoing transition, such that if the guard expression is false, the transition is disabled.

The following example illustrates how guards can be applied to transitions coming into or out of a Junction

pseudo-state.



Toolbox Icon



Learn More:

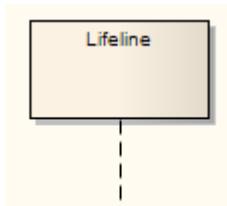
- [State Machine Diagram](#) ^[815]
- [Pseudo-states](#) ^[820]
- [Fork](#) ^[900]
- [Join](#) ^[901]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

... junction vertices are semantic-free vertices that are used to chain together multiple transitions. They are used to construct compound transition paths between states. For example, a junction can be used to converge multiple incoming transitions into a single outgoing transition representing a shared transition path (this is known as a merge). Conversely, they can be used to split an incoming transition into multiple outgoing transition segments with different guard conditions. This realizes a static conditional branch. (In the latter case, outgoing transitions whose guard conditions evaluate to false are disabled. A predefined guard denoted "else" may be defined for at most one outgoing transition. This transition is enabled if all the guards labeling the other transitions are false.) Static conditional branches are distinct from dynamic conditional branches that are realized by choice vertices.

6.4.1.25 Lifeline



Description

A Lifeline is an individual participant in an interaction (that is, Lifelines cannot have multiplicity). A Lifeline represents a distinct connectable element. To specify that representation within Enterprise Architect, right-click on the Lifeline and select the Advanced | Instance Classifier context menu option. The Select <Item> dialog displays which you use to locate the required project classifiers.

Lifelines are available in Sequence diagrams. There are different Lifeline elements for Timing diagrams (State Lifeline and Value Lifeline); however, although the representation differs between the two diagram types, the meaning of the Lifeline is the same.

Toolbox Icon



Learn More:

- [Instance Classifier](#)^[690]
- [Sequence Diagram](#)^[849]
- [Timing diagram](#)^[830]
- [State Lifeline](#)^[923]
- [Value Lifeline](#)^[938]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p.489*) states:

A lifeline represents an individual participant in the Interaction. While Parts and StructuralFeatures may have multiplicity greater than 1, Lifelines represent only one interacting entity.

Lifeline is a specialization of NamedElement.

If the referenced ConnectableElement is multivalued (i.e. has a multiplicity > 1), then the Lifeline may have an expression (the 'selector') that specifies which particular part is represented by this Lifeline. If the selector is omitted this means that an arbitrary representative of the multivalued ConnectableElement is chosen.

6.4.1.26 Merge



Description

A Merge Node brings together a number of alternative flow paths in Activity, Analysis and Interaction Overview diagrams. For example, if a Decision is used after a Fork, the two flows coming out of the Decision must be merged into one before going to a Join; otherwise, the Join waits for both flows, only one of which arrives.

A Merge Node has multiple incoming edges and a single outgoing edge. The edges coming into and out of a Merge Node must be either all object flows or all control flows.

Toolbox Icon



Learn More:

- [Activity Diagram](#) ^[81]
- [Analysis Diagram](#) ^[1187]
- [Interaction Overview Diagram](#) ^[86]
- [Decision](#) ^[886]
- [Fork](#) ^[900]
- [Join](#) ^[90]
- [Object flow](#) ^[1003]
- [Control flow](#) ^[977]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 387*) states:

A merge node is a control node that brings together multiple alternate flows. It is not used to synchronize concurrent flows but to accept one among several alternate flows.

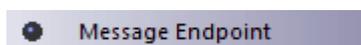
6.4.1.27 Message Endpoint



Description

A Message Endpoint element defines the termination of a State or Value Lifeline in a Timing diagram.

Toolbox Icon



Learn More:

- [Timing Diagram](#) ^[830]
- [State Lifeline](#) ^[923]

- [Value Lifeline](#)^[938]

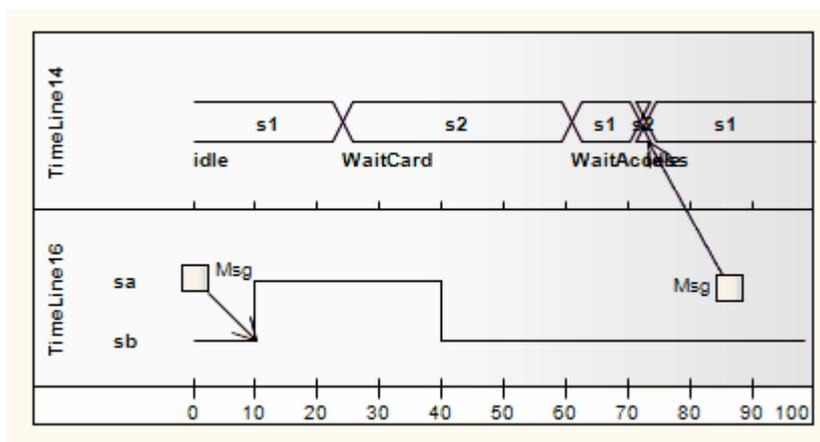
6.4.1.28 Message Label



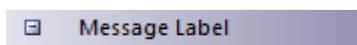
Description

A Message Label is an alternative way of denoting Messages between Lifelines, which is useful for 'uncluttering' Timing diagrams strewn with messages. To indicate a Message between Lifelines, draw a connector from the source Lifeline into a Message Label. Next, draw a connector from another Message Label to the target Lifeline. Note that the label names must match to reflect that the message occurs between the two Message Labels.

The following diagram illustrates how Message Labels are used to construct a message between Lifelines.



Toolbox Icon



Learn More:

- [Timing diagram](#)^[830]
- [Messages](#)^[999]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 518*) states:

Labels are only notational shorthands used to prevent cluttering of the diagrams with a number of messages crisscrossing the diagram between Lifelines that are far apart. The labels denote that a Message may be disrupted by introducing labels with the same name.

6.4.1.29 Note



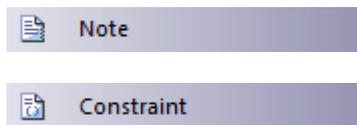
Description

A Note element is a textual annotation that can be attached to a set of elements of any other type. The attachment is created separately, using a Notelink connector. Both Note and Notelink are available in any Enterprise Architect diagram, through the Common pages of the Toolbox.

A Note is also called a Comment.

A Constraint is a form of Note, identifying a constraint on other elements. As for a Note, you can connect the Constraint element to other elements using a Notelink connector. This element is just a means of documenting the fact that there are constraints; it has no impact on the other elements. You define the types of constraint in the project reference data, apply them to the element in the element Properties dialog, and manage them through the Scenarios & Requirements window.

Toolbox Icon



Learn More:

- [Notelink Connector](#)^[1003]
- [Constraint Types](#)^[784]
- [Scenarios & Requirements Window](#)^[689]

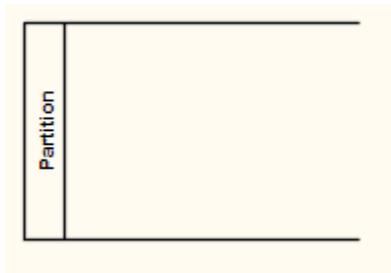
OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 59*) states:

A comment gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeler.

A comment can be owned by any element.

6.4.1.30 Partition



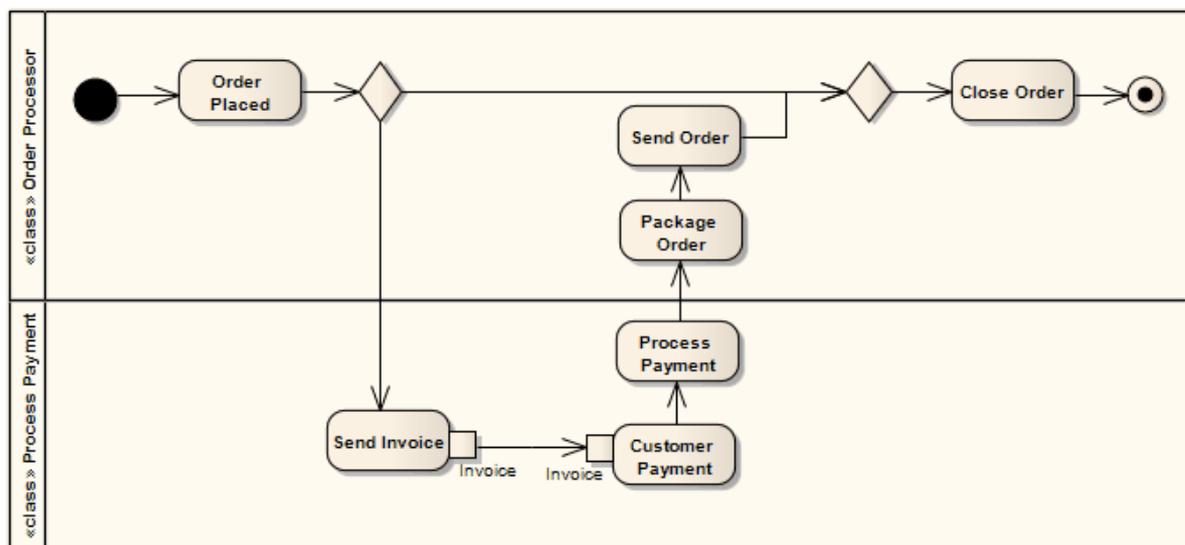
Description

Enterprise Architect supports two types of Activity Partition:

- The Activity Partition feature, which is used to logically organize an Activity element
- The Activity Partition element, described in this topic, which is used to logically organize an Activity diagram.

In effect, these are the same. They partition the Actions of the Activity without affecting the token flow, helping to structure the view or parts of the Activity.

The following example depicts the partitioning between the Classes Process Payment and Order Processor.



The Partition orientation defaults to horizontal. To turn it into a vertical Partition, right-click on it and select the Advanced | Vertical Partition context menu option.

You can neatly align and join the Activity Partitions on a diagram using the element context menu Dockable option. For Partitions, the option defaults to selected.

Toolbox Icon



Learn More:

- [Activity Diagram](#)^[811]
- [Activity Element](#)^[873]
- [Activity Partition Feature](#)^[876]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) states:

Partitions divide the nodes and edges to constrain and show a view of the contained nodes. Partitions can share contents. They often correspond to organizational units in a business model. They may be used to allocate characteristics or resources among the nodes of an activity.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) also states:

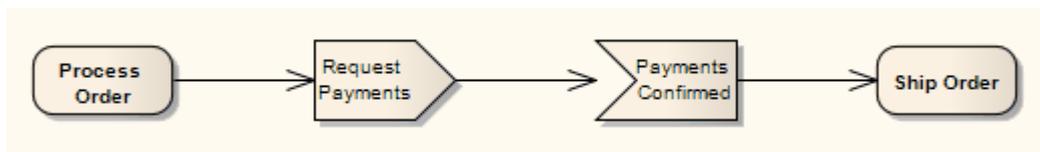
An activity partition is a kind of activity group for identifying actions that have some characteristic in common.

6.4.1.31 Receive**Description**

A Receive element is used to define the acceptance or receipt of a request, in an Activity diagram. Movement from a Receive element occurs only once receipt is fulfilled according to its specification. The Receive element comes in two forms:

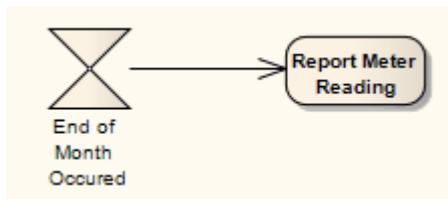
- *Accept Event Action* element (pennant shape)
- *Accept Time Event Action* element (hourglass shape)

The following example reflects a payment process on an order. Upon receiving the payment (from Request Payments, a Send element), the payment is confirmed and the flow continues to ship the order.



See *UML Superstructure Specification, v2.1.1, figure 12.26, p. 312*.

To depict an Accept Time Event, use the standard Receive element from the **Toolbox**. Right-click on this element, and select the **Advanced | Accept Time Event context** menu option. The following example shows the hourglass-shaped Accept Time Event Action:



See *UML Superstructure Specification, v2.1.1, figure 12.27, p. 312.*

Toolbox Icon



Learn More:

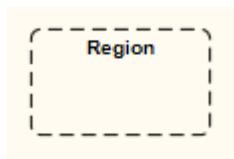
- [Activity Diagram](#)^[81]
- [Send Element](#)^[91]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 239*) states:

AcceptEventAction is an action that waits for the occurrence of an event meeting specified conditions.

6.4.1.32 Region



Description

Enterprise Architect supports two types of *Region* element:

- Expansion Region
- Interruptible Activity Region

When you add a *Region* element to an Activity diagram, a dialog prompt appears. Use this to specify the *Region* type.

Toolbox Icon



Learn More:

- [Activity Diagram](#)^[81]
- [Expansion Region](#)^[89]

- [Interruptible Activity Region](#)^[90]

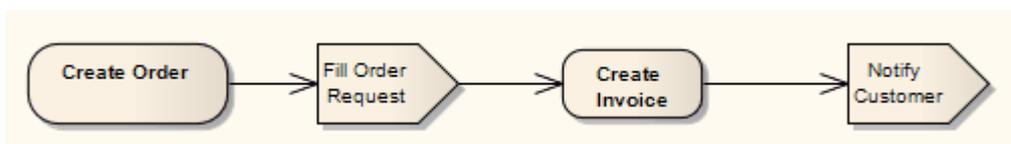
6.4.1.33 Send



Description:

The Send element is used to depict the action of sending a signal, in an Activity diagram. It is the opposite of a Receive element.

The following example shows an order being processed, where a signal is sent to fill the processed order and, upon creation of the resulting invoice, a notification is sent to the customer.



See *UML Superstructure Specification, v2.1.1, figure 12.132, p. 408.*

Toolbox Icon:



Learn More:

- [Activity Diagram](#)^[81]
- [Receive](#)^[91]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 285*) states:

SendObjectAction is an action that transmits an object to the target object, where it may invoke behavior such as the firing of state machine transitions or the execution of an activity. The value of the object is available to the execution of invoked behaviors. The requestor continues execution immediately. Any reply message is ignored and is not transmitted to the requestor.

6.4.1.34 State



Description

A *State* represents a situation where some invariant condition holds; this condition can be static (waiting for an event) or dynamic (performing a set of activities). State modeling is usually related to Classes, and describes the enableable states a Class or element can be in and the transitions that enable the element to move there. There are two types of State: *Simple States* and *Composite States*, both created from the State element from the **Toolbox**.

Furthermore, there are pseudo-states, resembling some aspect of a State but with a pre-defined implication. Pseudo-states model complex transitional paths, and classify common State Machine behavior.

You can define entry, internal and exit actions for a State using operations.

If a State element has features such as attributes or operations, the depiction of the element in a diagram has a line under the element name. This line persists if the features are hidden. The line also displays if the **Show State Compartment** checkbox is selected on the **Objects** page of the **Options** dialog (**Tools | Options | Objects**).



Toolbox Icon



See Also:

- [Composite State](#) ^[919]

Learn More:

- [State Machine Diagram](#) ^[815]
- [Class](#) ^[941]
- [Pseudo-states](#) ^[820]
- [Operations](#) ^[708]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 546*) states:

A state models a situation during which some (usually implicit) invariant condition holds. The invariant may represent a static situation such as an object waiting for some external event to occur. However, it can also model dynamic conditions such as the process of performing some activity (i.e., the model element under consideration enters the state when the activity commences and leaves it as soon as the activity is completed).

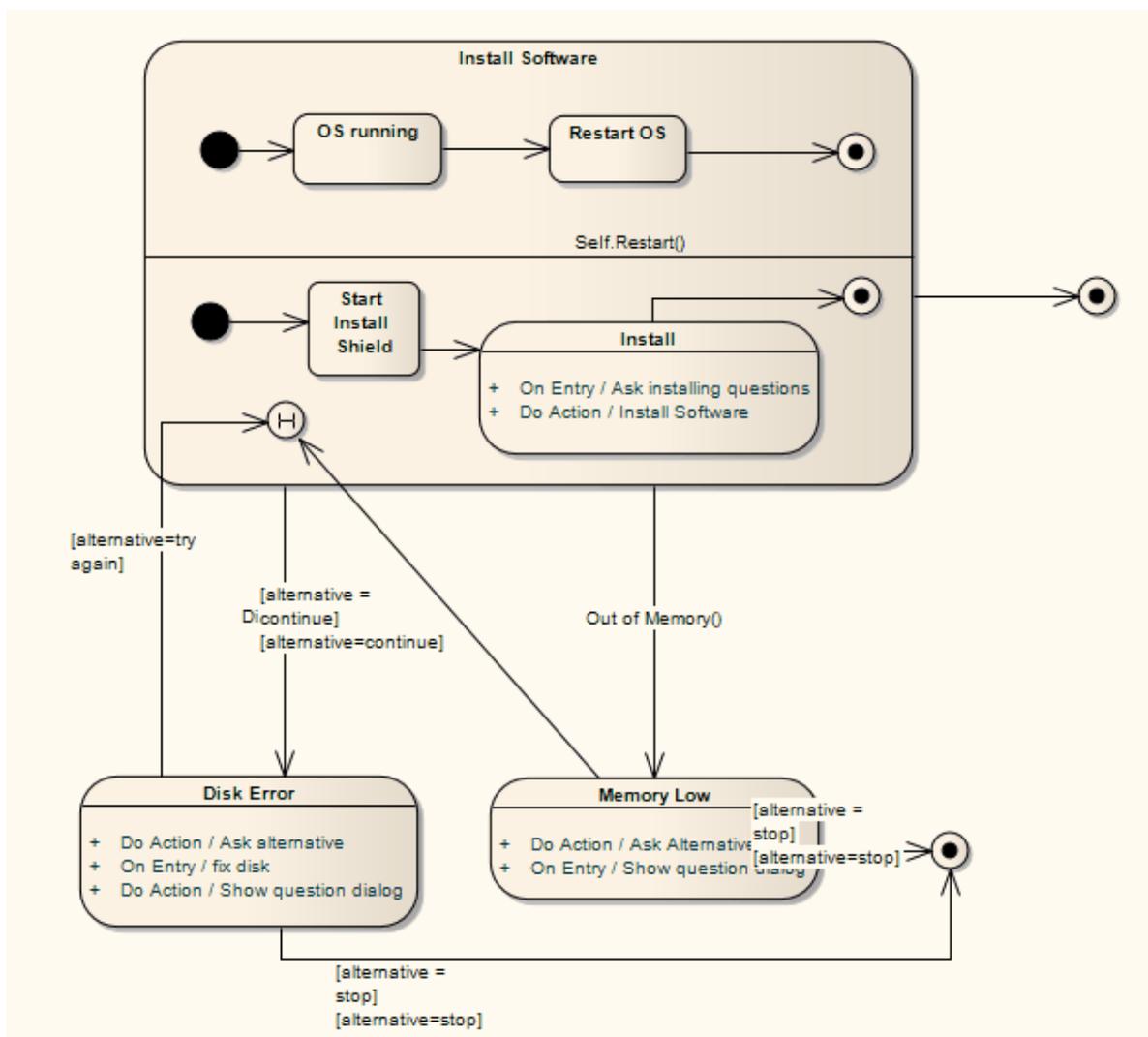
6.4.1.34.1 Composite State

Description

Composite States are composed within the State Machine diagram by expanding a State element, adding Regions if applicable, and dragging further State elements, related elements and connectors within its boundaries. The internal State elements are then referred to as Sub-states.

(You can also define a State element, as with many other types of element, as a composite element; this then has a hyperlink to a child diagram that can be another State Machine diagram or other type of diagram elsewhere in the model.)

Composite States can be orthogonal, if Regions are created. If a Composite State is orthogonal, its entry denotes that a single Sub-state is concurrently active in all Regions. The hierarchical nesting of Composite States, coupled with Region use, generates a situation of multiple States concurrently active; this situation is referred to as the active State configuration.



Learn More:

- [State Machine Diagram](#)^[815]
- [State Element](#)^[917]

- [Regions](#)^[820]
- [Composite Element](#)^[647]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 478*) states:

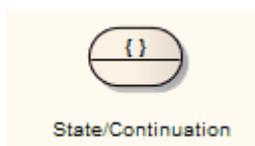
A composite state either contains one region or is decomposed into two or more orthogonal regions. Each region has a set of mutually exclusive disjoint subvertices and a set of transitions. A given state may only be decomposed in one of these two ways.

Any state enclosed within a region of a composite state is called a substate of that composite state. It is called a direct substate when it is not contained by any other state; otherwise it is referred to as an indirect substate.

Each region of a composite state may have an initial pseudostate and a final state. A transition to the enclosing state represents a transition to the initial pseudostate in each region. A newly-created object takes its topmost default transitions, originating from the topmost initial pseudostates of each region.

A transition to a final state represents the completion of activity in the enclosing region. Completion of activity in all orthogonal regions represents completion of activity by the enclosing state and triggers a completion event on the enclosing state. Completion of the topmost regions of an object corresponds to its termination.

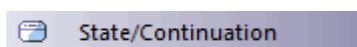
6.4.1.35 State/Continuation



Description

The State/Continuation element serves two different purposes for Interaction (Sequence) diagrams, as State Invariants and Continuations. Enterprise Architect prompts you to identify the purpose when you create the element.

Toolbox Icon



See Also:

- [Continuation](#)^[921]
- [State Invariant](#)^[922]

Learn More:

- [Sequence Diagram](#)^[849]

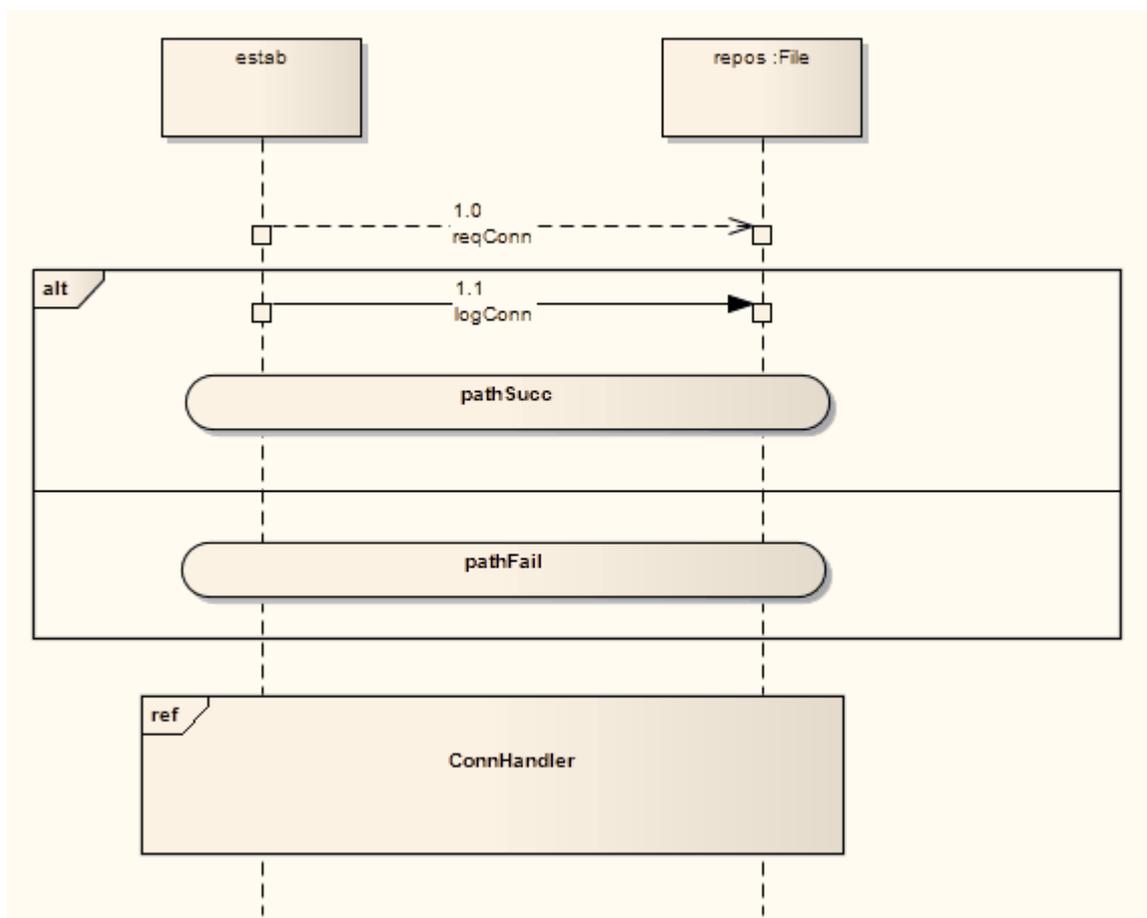
6.4.1.35.1 Continuation

Description

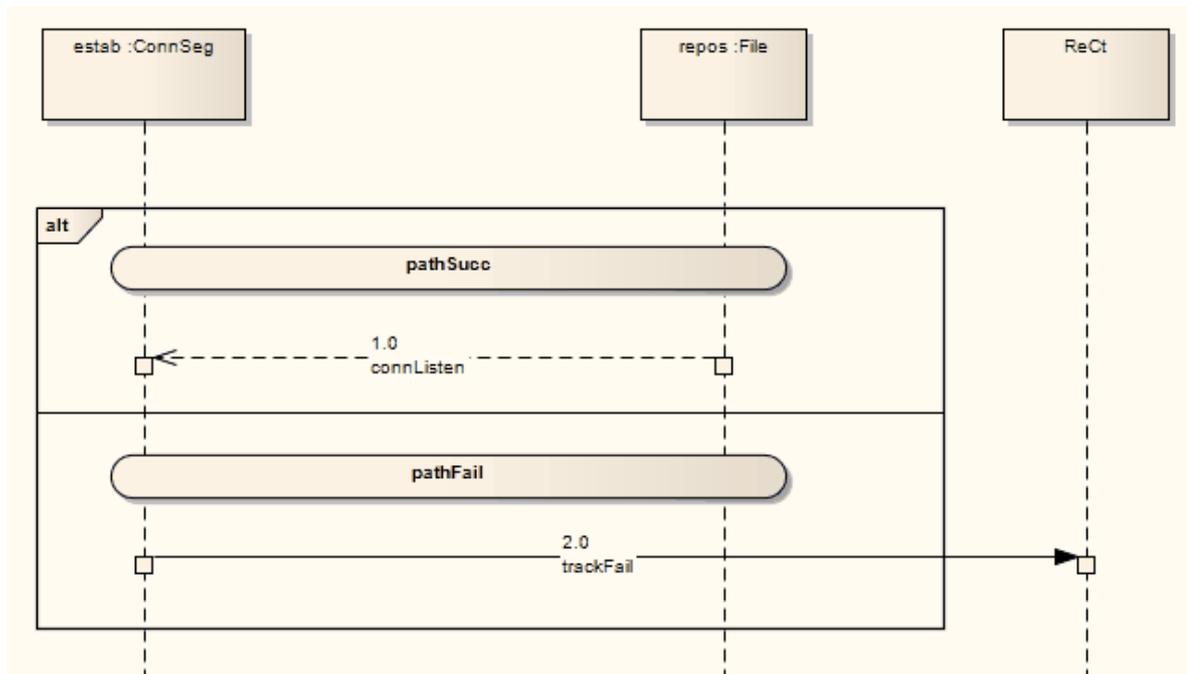
A Continuation is used in seq and alt Combined Fragments, to indicate the branches of continuation an operand follows. To indicate a continuation, end an operand with a Continuation, and indicate the continuation branch with a matching Continuation (same name) preceding the Interaction Fragment.

You create a Continuation by dragging the State/Continuation element onto the diagram from the **Interaction Elements** page of the **Toolbox**.

For the following continuation example, an alt Combined Fragment has Continuations pathSucc and pathFail. These Continuations are located within the Interaction Occurrence ConnHandler, which has subsequent events based on the continuation.



The following diagram shows the interaction referenced by the *Interaction Occurrence*.

**Learn More:**

- [Combined Fragments](#)^[880]
- [State/Continuation Element](#)^[920]
- [Interaction Occurrence](#)^[906]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 474*) states:

A Continuation is a syntactic way to define continuations of different branches of an Alternative CombinedFragment. Continuation is intuitively similar to labels representing intermediate points in a flow of control.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 474*) also states:

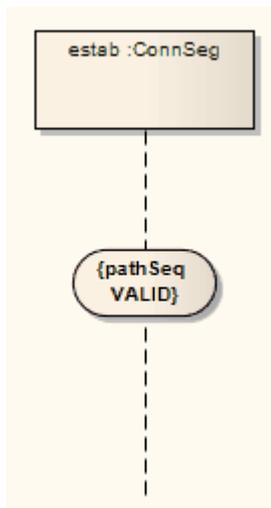
Continuations have semantics only in connection with Alternative CombinedFragments and (weak) sequencing.

If an InteractionOperand of an Alternative CombinedFragment ends in a Continuation with name (say) X, only InteractionFragments starting with the Continuation X (or no continuation at all) can be appended.

6.4.1.35.2 State Invariant**Description**

A State Invariant is a condition applied to a Lifeline, which must be fulfilled for the Lifeline to exist. You create a State Invariant by dragging the State/Continuation element onto the diagram from the **Interaction Elements** page of the **Toolbox**.

The following diagram illustrates a State Invariant.



When a State Invariant is moved near to a Lifeline, it snaps to the center. If the sequence object is dragged left or right, the State Invariant moves with it.

Learn More:

- [Lifeline](#)^[910]
- [State/Continuation](#)^[920]

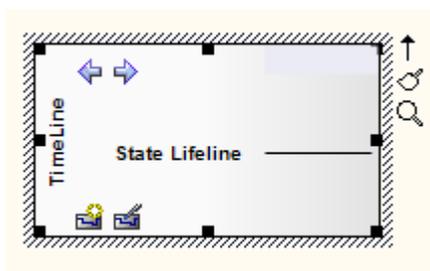
OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 502*) states:

A StateInvariant is a runtime constraint on the participants of the interaction. It may be used to specify a variety of different kinds of constraints, such as values of attributes or variables, internal or external states, and so on.

A StateInvariant is an InteractionFragment and it is placed on a Lifeline.

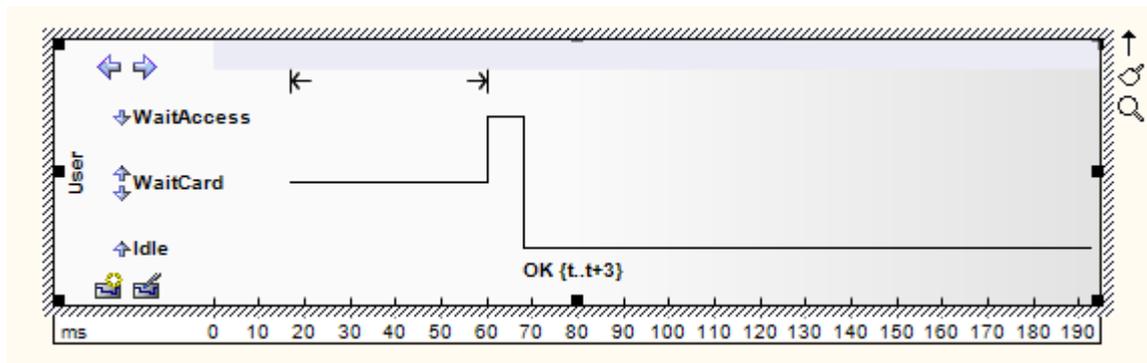
6.4.1.36 State Lifeline



Description

A Lifeline is the path an object takes across a measure of time, as indicated by the x-axis. There are two sorts: State Lifelines (defined here) and Value Lifelines, both used in Timing diagrams.

A State Lifeline follows discrete transitions between states, which are defined along the y-axis of the timeline. Any transition has optional attributes of timing constraints, duration constraints and observations. An example of a State Lifeline is shown below:



See *UML Superstructure Specification, v2.1.1, figure 14.29, p. 519.*

A State Lifeline consists of a set of transition points. Each transition point can be defined with the following properties:

Property	Description
At time	Specifies the starting time for a change of state.
Transition to	Indicates the state to which the lifeline changes.
Event	Describes the occurring event.
Timing constraints	Refers to the time taken for a state to change within a lifeline, or the time taken to transmit a message (e.g. t..t+3).
Timing observations	Provides information on the time of a state change or sent message.
Duration constraints	Pertains to a lifeline's period at a particular state. The constraint could be instigated by a change of state within a lifeline, or that lifeline's receipt of a message.
Duration observations	Indicates the interval of a lifeline at a particular state, begun from a change in state or message receipt.

In the example diagram above, the **OK** transition point has these properties:

Property	Value
At Time	18 ms
Transition to	Idle
Event	OK
Timing constraints	t..t+3
Timing observations	–
Duration constraints	–
Duration observations	–

Toolbox Icon

 State Lifeline**Learn More:**

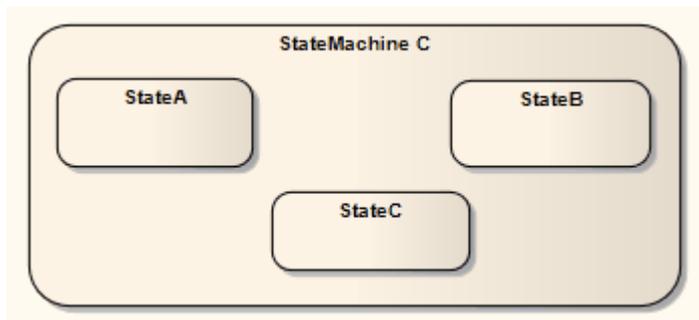
- [Timing Diagram](#) ^[830]
- [Value Lifelines](#) ^[938]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 518*) states:

This is the state of the classifier or attribute, or some testable condition, such as an discrete enumerable value.

It is also permissible to let the state-dimension be continuous as well as discrete. This is illustrative for scenarios where certain entities undergo continuous state changes, such as temperature or density.

6.4.1.37 State Machine**Description**

A State Machine element is a container for groups of related State elements. You can create sections of a State Machine diagram, showing the organization of the inter-related State elements, and enclose each section in a State Machine element. You can also create Regions on a State Machine element.

Toolbox Icon State Machine**Learn More:**

- [Regions](#) ^[820]

6.4.1.38 Structured Activity

Description

Structured Activity elements are used in Activity diagrams. A Structured Activity is an activity node that can have subordinate nodes as an independent *Activity Group*. No other Activities or their side effects should interfere with this Activity's processing.

Enterprise Architect provides two forms of Structured Activity - basic and specialized. It also applies the mechanism for creating Structured Activities to creating composite Activity elements quickly and simply.

The two basic Structured Activities are:

- Structured Activity Node - represents an ordered arrangement of executable Activity nodes (Actions, Decisions, Merges and so on) that can include branched and nested nodes; this is the base element from which the other types of Structured Activity are derived
- Sequential Node - represents a sequential arrangement of executable Activity nodes.

The two specialized Structured Activities are used to effectively model discreet patterns within an activity graph, defined in Clauses or Partitions:

- Conditional node - represents an arrangement of Actions and Activities where choice determines which Activities are performed
- Loop node - represents a sequence of Actions and Activities that are - or can be - repeated on the same object.

All four kinds of Structured Activity node are created as composite elements. However, for the Loop Node and Conditional Node elements you must create the child element structure on the parent diagram within the node element itself, as for a Composite State. You cannot develop the partitioned structure of the nodes on a child diagram. For this reason, the **Show Composite Diagram** facility is not available for the Loop Node and Conditional Node. It is also not available on the Structured Activity Node, as this is the base element for the Loop and Conditional Nodes. You can, however, use the two basic nodes as composite elements, and display the child diagram structure on the parent Sequential node.

When you create a Structured Activity, by selecting the icon from the **Activity** page of the **Toolbox**, the following choices are offered in a context menu:

- Loop Node
- Conditional Node
- Other

The first two options specifically create a Loop or Conditional Node. The **Other** option displays the **New Structured Activity** dialog, on which you can select to create any of the four nodes, or a simple Composite Activity element.

Toolbox Icon



See Also:

- [Structured Node](#)^[927]
- [Sequential Node](#)^[927]
- [Loop Node](#)^[928]
- [Conditional Node](#)^[929]

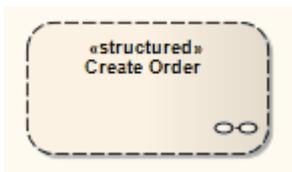
Learn More:

- [Activity Diagram](#) ^[81]
- [Activity](#) ^[873]
- [Composite Elements](#) ^[647]
- [Composite State](#) ^[919]

6.4.1.38.1 Structured Node

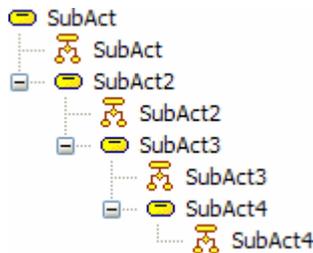
Description:

On a diagram, *Structured* Activity Nodes have broken borders and composite diagram icons, as shown below:



To display the Activity diagram represented by a Structured Activity Node element, double-click on the element.

Structured Activity Node elements can point to child diagrams that themselves contain or consist of Structured Activity elements; that is, the Structured Activity elements are nested, as shown in the section of **Project Browser** below.



OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 409*) states:

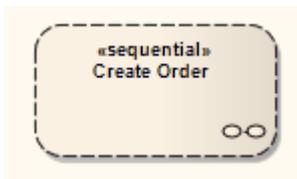
A structured activity node is an executable activity node that may have an expansion into subordinate nodes as an ActivityGroup. The subordinate nodes must belong to only one structured activity node, although they may be nested.

A structured activity node represents a structured portion of the activity that is not shared with any other structured node, except for nesting.

6.4.1.38.2 Sequential Node

Description:

On a diagram, *Sequential* Activity Nodes have broken borders and composite diagram icons, as shown below:



To display the Activity diagram represented by a Sequential Activity Node element, double-click on the element.

OMG UML Specification

Sequential Node

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 408*) states:

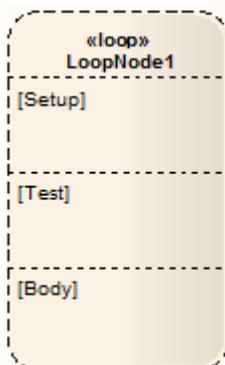
A sequence node is a structured activity node that executes its actions in order.

6.4.1.38.3 Loop Node

Description:

A *Loop Structured Activity Node* by default has *Setup*, *Test* and *Body* partitions. The Setup partition is executed once on entry to the Loop, and the Test and Body partitions are executed repeatedly until the Test produces a false value. The results of the final execution of the Test or Body are available after execution of the Loop is complete.

They are depicted on an Activity diagram as shown below:



You define the Loop nodes by dragging other Activity diagram elements from the **Toolbox** page into the appropriate partition of the element, and linking and organizing the structure as required. The elements are aligned on the top left of the partition, so that resizing the node maintains the organization of the structure within and between the partitions. If you try to shrink the node below the structure size, the node automatically defaults to the 'best fit' size.

When you create a Loop Node, the element **Properties** dialog displays. Much of this you can complete as for any other element. However, for the Loop Node the dialog also has a **Loop** tab.

Add an Action Pin for each of the **Decider**, **Loop Variable Input**, **Loop Variable**, **Body Output** and **Result** fields for the node, in each case clicking on the **Browse** or **Add** button to display the **Select Pins** dialog (a version of the **Select <Item>** dialog). The **Select ActionPin** dialog reveals only Input pins (**Loop Variable Input**) or Output pins as appropriate to the context. If the required Action Pin does not already exist, you can

click on the **Add New** button on the dialog to automatically create the Input pin or an Output pin under the appropriate parent node.

You can also check on the exact location of an existing Action Pin by right-clicking on the pin name and selecting the **Find in Project Browser** context menu option.

The **Nodes** panel, by default, lists the Actions and Activities contained in the **Setup** partition. Click on the **Body** or **Test** radio buttons to list the elements contained in the corresponding partitions. An element must be completely contained in a partition to be listed there - if it overlaps with the partition above in any way, it is treated as being part of that partition.

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, pp. 384-385*) states:

A loop node is a structured activity node that represents a loop with setup, test, and body sections.

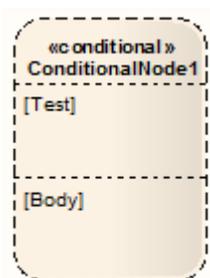
Each section is a well-nested subregion of the activity whose nodes follow any predecessors of the loop and precede any successors of the loop. The test section may precede or follow the body section. The setup section is executed once on entry to the loop, and the test and body sections are executed repeatedly until the test produces a false value. The results of the final execution of the test or body are available after completion of execution of the loop.

6.4.1.38.4 Conditional Node

Description:

A *Conditional* Structured Activity Node has a default *Clause* containing *Test* and *Body* partitions. Further Clauses can be added if required.

They are depicted on an Activity diagram as shown below:



You define Conditional nodes by dragging other Activity diagram elements from the **Toolbox** page into the appropriate partition of the element, and linking and organizing the structure as required. The elements are aligned on the top left of the partition, so that resizing the node maintains the organization of the structure within and between the partitions. If you try to shrink the node below the structure size, the node automatically defaults to the 'best fit' size.

When you create a Conditional Node, the element **Properties** dialog displays. Much of this you can complete as for any other element. However, for the Conditional Node the dialog shows an additional page.

Add an Action Pin as the **Result** for the node, clicking on the **Add** button to display the **Select Pins** dialog (a version of the **Select <Item>** dialog).

On creation, the Conditional Node automatically has one *Clause* containing a **Decider** and **Body Output**, and a *Test* partition and a *Body* partition. You can add further Clauses as required. For each Clause you also add an Action Pin for the **Decider** and for the **Body Output**. Click on the **Save** button to save the Clause definition.

The **Select Pin** dialog reveals only Output pins as appropriate to the context. If the required Action Pin does not already exist, you can click on the **Add New** button on the dialog to automatically create an Output pin under the appropriate parent node.

For the **Result** and **Body Output** entries, you can check on the exact location of each Action Pin by right-clicking on the entry and selecting the **Find in Project Browser** context menu option.

The **Nodes** panel, by default, lists the Actions and Activities contained in the Test partition. Click on the **Body** radio button to list the elements contained in the Body partition. An element must be completely contained in the Body partition to be listed there - if it overlaps with the Test partition in any way, it is treated as being part of the Test partition.

Add or Remove Clauses

To add another Clause, click on the **Add** button underneath the **Clause(s)** list. This inserts a new Clause in the list, and identifies which is the preceding, or Predecessor, Clause and (if appropriate) which is the following, or Successor, Clause. The remaining fields in the **Clause(s)** panel are cleared to enable you to add **Decider** and **Body Output** Action Pins. New Test and Body partitions are immediately added to the element on the diagram, and you can populate these partitions with Activity elements, which are then identified in the **Nodes** panel.

To remove a Clause, highlight it in the list and click on the **Delete** button. This immediately removes the Clause's corresponding partitions from the diagram, along with all their contained Activity elements. Removing a Clause from between two other Clauses adjusts the numerical order; for example, if Clause 2 is removed from between Clause 1 and Clause 3, Clause 3 is renamed as Clause 2, and any further Clauses are also moved up one place.

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p.355*) states:

A conditional node is a structured activity node that represents an exclusive choice among some number of alternatives.

A conditional node consists of one or more clauses. Each clause consists of a test section and a body section. When the conditional node begins execution, the test sections of the clauses are executed. If one or more test sections yield a true value, one of the corresponding body sections will be executed. If more than one test section yields a true value, only one body section will be executed. The choice is nondeterministic unless the test sequence of clauses is specified. If no test section yields a true value, then no body section is executed; this may be a semantic error if output values are expected from the conditional node.

6.4.1.39 Synch



Description

A Synch state is useful for indicating that concurrent paths of a State Machine are synchronized. After bringing the paths to a synch state, the emerging transition indicates unison.

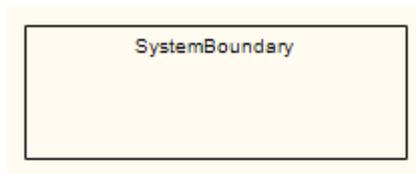
Toolbox Icon

 Synchronisation

Learn More:

- [State Machine Diagram](#) ^[815]

6.4.1.40 System Boundary



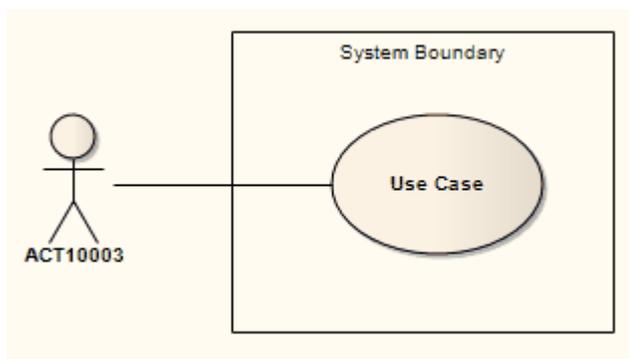
Description

A *System Boundary* element is a non-UML element used to define *conceptual* boundaries. You can use System Boundaries to help group logically related elements (from a visual perspective, not as part of the UML model).

In the *UML Superstructure Specification, v2.1.1*, System Boundaries are described in the sections on Use Cases, because the System Boundary is often used to indicate the *application* of a Use Case to another entity. In this context, the System Boundary:

- encloses the Use Case, and
- is associated with a classifier such as a Class, Component or Sub-system (Actor) through the **Select <Item>** dialog.

By associating the System Boundary - and not the Use Case - with the classifier, the classifier is linked to the Use Case as a *user*, but not as an *owner*.



You can also define a Use Case as the classifier of a System Boundary element, to link the elements enclosed in the System Boundary (such as parts of an Activity diagram) to their representation in a logical Use Case. See http://www.sparxsystems.com.au/resources/map_uc.html.

The following properties of a System Boundary can be set: the name, the border style, and the number of horizontal or vertical swim lanes.

A System Boundary element can be marked as *Selectable*, using the element's context menu. When not

selectable, you can click within the System Boundary space without activating or selecting the Boundary itself. This is useful when you have many elements within the Boundary and the Boundary makes their selection difficult.

Notes:

- A System Boundary is the basis for the Image element, which enables you to add icons or backgrounds to a diagram, automatically displaying the Image Manager dialog from which to select the appropriate image.

Toolbox Icon



Learn More:

- [Use Cases](#)^[935]
- [Instance Classifier](#)^[690]
- [Boundary Element Settings](#)^[932]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 594*) states:

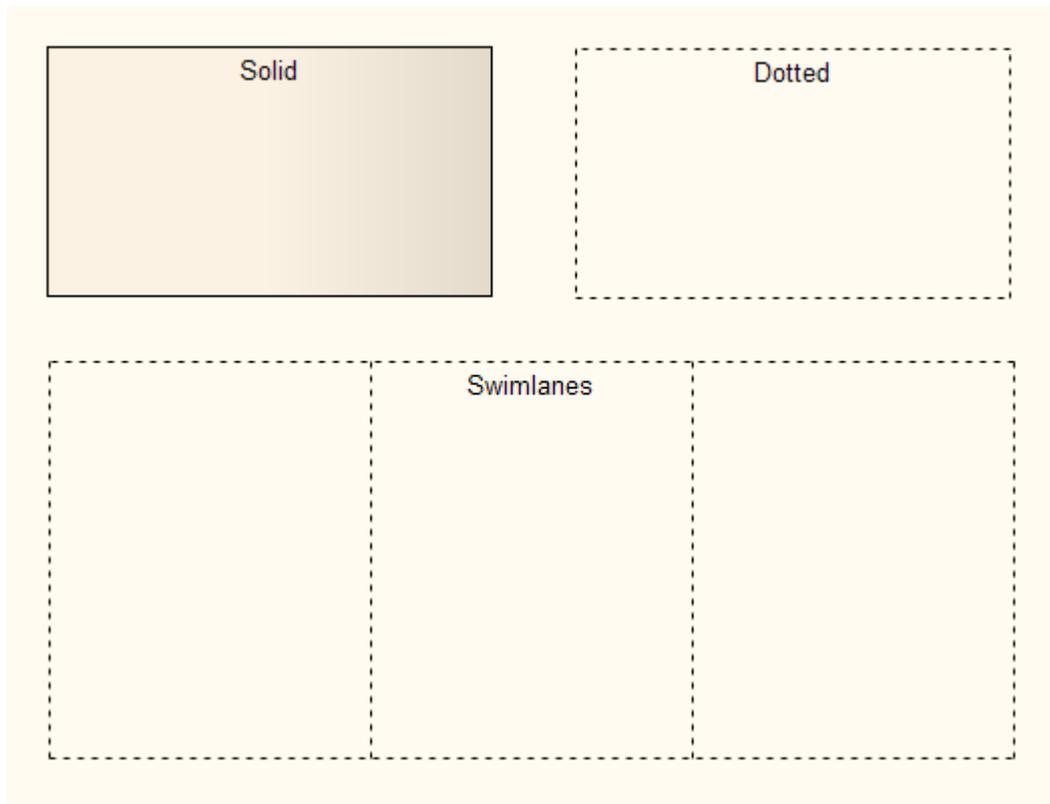
If a subject (or system boundary) is displayed, the Use Case ellipse is visually located inside the system boundary rectangle. Note that this does not necessarily mean that the subject classifier owns the contained Use Cases, but merely that the Use Case applies to that classifier.

6.4.1.40.1 Boundary Element Settings

Configure Boundary Elements

Boundary elements can be configured to display in different ways. The main differences are:

- Solid border
- Dotted border
- With horizontal or vertical 'swim lanes'; swim lanes are used to group elements in a vertical or horizontal context (for example, Client, Application and Database tiers could be represented in swim lanes).



6.4.1.41 Terminate



Description

The Terminate pseudo-state indicates that upon entry of its pseudo-state, the State Machine's execution ends.

Toolbox Icon



Learn More:

- [State Machine Diagram](#) ^[815]
- [Pseudo-state](#) ^[820]

6.4.1.42 Trigger



Description

A Trigger indicates an event that initiates an action (and might arise from completion of a previous action). You initially define a Trigger in one of four ways:

- As a property of a Transition relationship
- As a property of an Accept Event Action (on the **Triggers** tab of the element **Properties** dialog)
- As an event in a State Machine Table
- Directly, as a Trigger element, through the **New Element** dialog.

When you save the Trigger, it is added to the list of elements for the parent package in the **Project Browser**. You can then right-click on it and select the **Properties** context menu option to view and, if required, edit its properties as an element rather than as a property itself. Triggers created as events remain as Event elements, whilst Triggers created in other ways are Trigger elements, with a **Triggers** tab in the **Properties** dialog.

Field	Action	See also
Type	<p>If necessary, edit the type of trigger:</p> <ul style="list-style-type: none"> • Call - specifies that the event is a CallEvent, which sends a message to the associated object by invoking an operation. • Change - specifies that the event is a ChangeEvent, which indicates that the transition is the result of a change in value of an attribute. • Signal - specifies that the event is a SignalEvent, which corresponds to the receipt of an asynchronous signal instance. • Time - corresponds to a TimeEvent; which specifies a moment in time. 	
Specification	<p>Either type in the event instigating the Trigger, or click on the (...) button and select the event (depending on the Type value).</p>	
Ports	<p>Click on the Add button and select the appropriate Port from the Select Port dialog.</p> <p>Notes:</p> <ul style="list-style-type: none"> • To create new Ports using the Select Port dialog, the Trigger should be created as a child of a Class or Component element. • To add several Ports at once, press (Ctrl) as you select each Port. • To check the exact location of a Port, right-click on the Port name and select the Find in Project Browser context menu option. 	Select Port Dialog <small>690</small>

You can also drag the Trigger element onto another diagram, although there are limited uses for the element in that context.

This element is not the same as a Trigger Operation, which is an operation automatically executed as a

result of the modification of data in a database.

Learn More:

- [Action](#)^[865]
- [Transition Relationship](#)^[1013]
- [Insert Trigger](#)^[825]
- [Trigger Operation](#)^[1365]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 456*) states:

Events may cause execution of behavior (e.g., the execution of the effect activity of a transition in a state machine). A trigger specifies the event that may trigger a behavior execution as well as any constraints on the event to filter out events not of interest.

6.4.1.43 Use Case

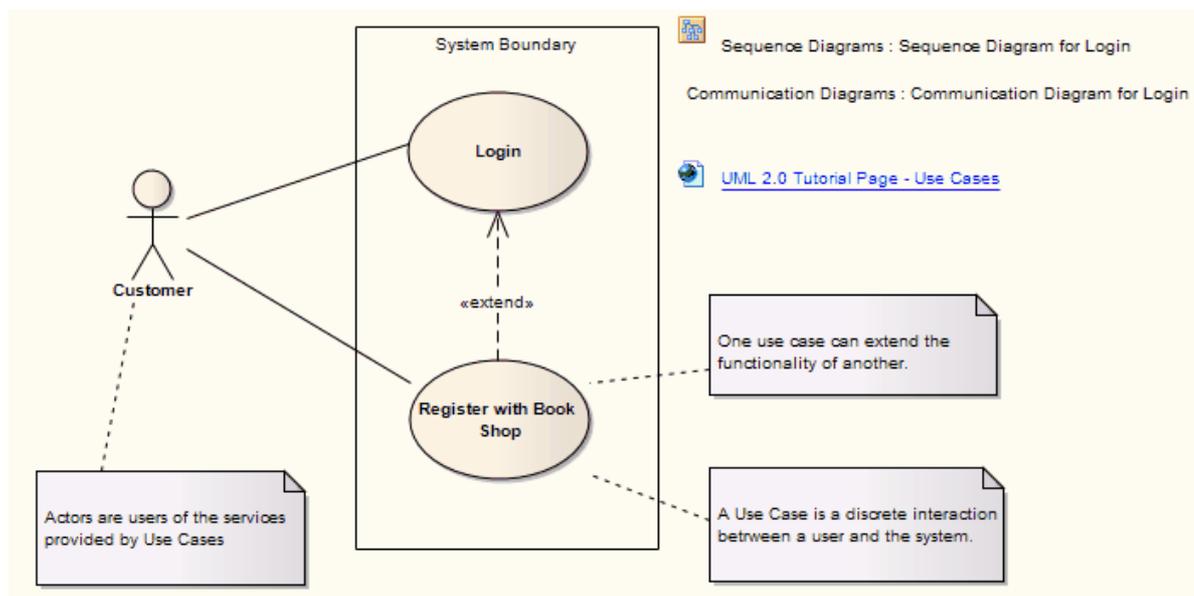


Description

A *Use Case* is a UML modeling element that describes how a user of the proposed system interacts with the system to perform a discrete unit of work. It describes and signifies a single interaction over time that has meaning for the end user (person, machine or other system), and is required to leave the system in a complete state: the interaction either completed or rolled back to the initial state. A Use Case:

- Typically has requirements and constraints that describe the essential features and rules under which it operates
- Can have an associated Sequence diagram illustrating behavior over time; who does what to whom, and when
- Typically has scenarios associated with it that describe the work flow over time that produces the end result; alternative work flows (for example, to capture exceptions) are also enabled.

The following is an example Use Case model:



If extending a Use Case, you can specify the points of extension with Use Case Extension Points. To display the attributes, operations or constraints of a Use Case on a diagram, use Rectangle Notation.

Enterprise Architect also provides two stereotyped Use Cases - the Test Case and the Business Use Case.

Toolbox Icon



See Also:

- [Use Case Extension Points](#) ^[937]
- [Rectangle Notation](#) ^[938]

Learn More:

- [Sequence Diagram](#) ^[849]
- [Test Case](#) ^[1674]
- [Business Use Case](#) ^[1191]

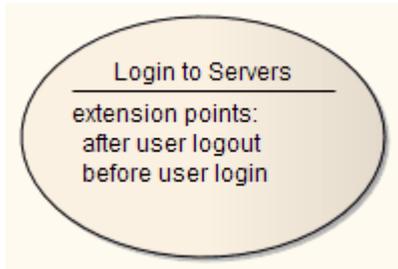
OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 592*) states:

A UseCase is a kind of behavior classifier that represents a declaration of an offered behavior. Each Use Case specifies some behavior, possibly including variants, that the subject can perform in collaboration with one or more actors.

6.4.1.43.1 Use Case Extension Points

Use *extension points* to specify the point of an extended Use Case where an extending Use Case's behavior should be inserted. The specification text can be informal or precise to define the location of the extension point.



How to:

To work with extension points, follow the steps below:

Step	Action
1	Right-click on the Use Case element The context menu displays
2	Select the Advanced Edit Extension Points... menu option The Use Case Extension Points dialog displays, listing defined points for that Use Case
3	Select an extension point in the list and click on the appropriate button to edit or remove the extension point, or to add a new one

Notes:

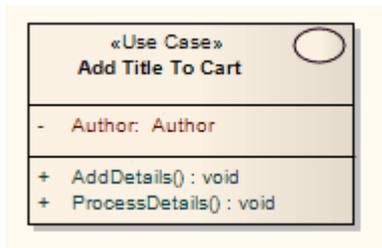
- Conditions to apply the extending Use Case, and the extension point to use, should be attached as a note to the extend relationship

Learn More:

- [Use Case](#)^[935]

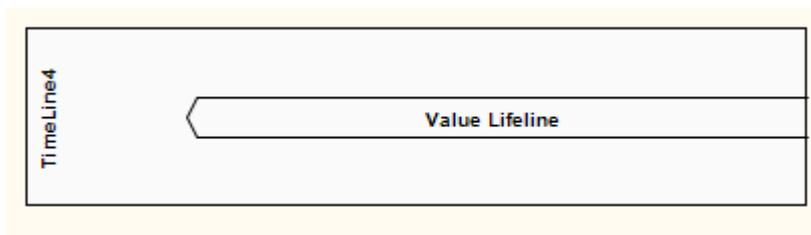
6.4.1.43.2 Rectangle Notation

You can display various shaped elements, such as an Interface, Use Case or Actor, using *rectangle notation*. This displays the element as a rectangle, with an icon of the 'normal' shape in the top right-hand corner. Any attributes, operations or constraints belonging to the element are shown, in the same style as a Class.



To show an element using rectangle notation, right-click on the element on the diagram and select the **Advanced | Use Rectangle Notation** context menu option. This setting only applies to the selected element, and can be toggled on and off either by deselecting the context menu option or by selecting the reciprocal option such as **Use Circle Notation** or **Use Actor Notation**.

6.4.1.44 Value Lifeline

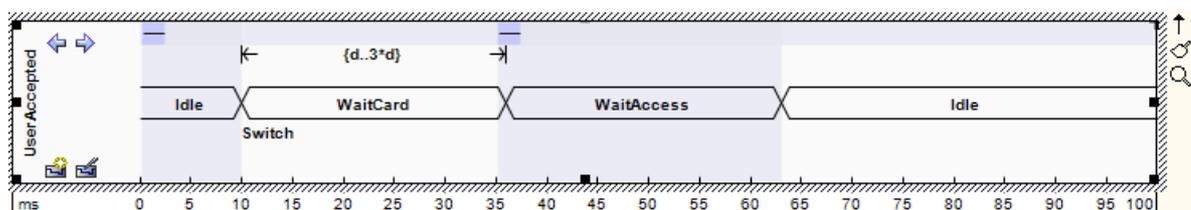


Description

A Lifeline is the path an object takes across a measure of time, indicated by the x-axis. There are two sorts: Value Lifelines (defined here) and State Lifelines, both used in Timing diagrams.

A Value Lifeline shows the Lifeline's state across the diagram, with parallel lines indicating a steady state. A cross between the lines indicates a transition or change in state.

An example of a Value Lifeline is shown below:



See *UML Superstructure Specification, v2.1.1, Figure 14.30, p. 520*.

A Value Lifeline consists of a set of transition points. Each transition point can be defined with the following properties:

Property	Description
At time	Specifies the starting time for a change of state.
Transition to	Indicates the state to which the Lifeline is to change.
Event	Describes the occurring event.
Timing constraints	Refers to the time taken for a state to change within a Lifeline, or the time taken to transmit a message.
Timing observations	Provides information on the time of a state change or sent message.
Duration constraints	Pertains to a Lifeline's period at a particular state. The constraint could be instigated by a change of state within a Lifeline, or that Lifeline's receipt of a message.
Duration observations	Indicates the interval of a Lifeline at a particular state, begun from a change in state or message receipt.

In the example diagram above, the **10ms** transition point has these properties:

Property	Text
At Time	10ms
Transition to	Waitcard
Event	Switch
Timing constraints	–
Timing observations	–
Duration constraints	d..3*d
Duration observations	–

Toolbox Icon



Learn More:

- [Timing Diagrams](#)^[830]
- [State Lifelines](#)^[923]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 518*) states:

Shows the value of the connectable element as a function of time. Value is explicitly denoted as text. Crossing reflects the event where the value changed.

6.4.2 Structural Diagram Elements

The following elements are used in UML Structural Diagrams. For more information on using each element, click on the element name in this list:

- [Actor](#)^[87], [Artifact](#)^[94]
- [Class](#)^[94], [Collaboration](#)^[94], [Collaboration Use](#)^[94], [Component](#)^[94]
- [Data Type](#)^[94], [Deployment Specification](#)^[94], [Document Artifact](#)^[95]
- [Enumeration](#)^[95], [Execution Environment](#)^[95], [Expose Interface](#)^[95]
- [Information Item](#)^[95], [Interface](#)^[95]
- [Node](#)^[95], [Note](#)^[91]
- [Object](#)^[95]
- [Package](#)^[96], [Part](#)^[96], [Port](#)^[96], [Primitive](#)^[96]
- [Qualifiers](#)^[97]
- [Signal](#)^[96]

Learn More:

- [Structural Diagrams](#)^[79]

6.4.2.1 Artifact



Description

An *Artifact* is any physical piece of information used or produced by a system, represented in a Deployment Diagram.

Artifacts can have associated properties or operations, and can be instantiated or associated with other Artifacts. Examples of Artifacts include model files, source files, database tables, development deliverables or support documents. The files represented by the Artifact are listed on the **Files** tab of the element **Properties** dialog.

To open the files represented by the Artifact, click on the element on the diagram and press **(Ctrl+E)**. Each file is opened either on a separate tab in the **Diagram View** workspace (if the file can be opened within Enterprise Architect) or in the default Windows viewer/editor for the file type (if the file cannot be opened within Enterprise Architect).

Files can also be launched individually from the **Files** tab (opening in the Windows default editor), as for elements of any other type that have associated files.

Toolbox Icon



See Also:

- [Create Artifact For External File](#)^[94]

Learn More:

- [Deployment Diagram](#) ^[804]
- [Associated Files](#) ^[687]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 201*) states:

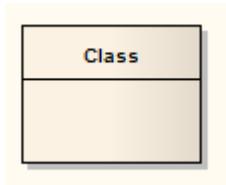
An Artifact defined by the user represents a concrete element in the physical world. A particular instance (or 'copy') of an artifact is deployed to a node instance. Artifacts may have composition associations to other artifacts that are nested within it. For instance, a deployment descriptor artifact for a component may be contained within the artifact that implements that component. In that way, the component and its descriptor are deployed to a node instance as one artifact instance.

6.4.2.1.1 Create Artifact For External File

You can create an Artifact element on a diagram for an external file, by clicking on the file in a file list (such as Windows Explorer) or on your Desktop and dragging it onto the diagram. A short context menu displays with two options - **Hyperlink** and **Artifact**.

Click on the **Artifact** option to create the element on the diagram. The **Properties** dialog displays, and you can define the name or other properties as required. Click on the **OK** button, and then open the **Properties** dialog again and click on the **Files** tab. The file pathname is listed in the **Files** panel.

6.4.2.2 Class



Description

A Class is a representation of objects that reflects their structure and behavior within the system. It is a template from which actual running instances are created, although a Class can be defined either to control its own execution or as a template or parameterized Class that specifies parameters that must be defined by any binding Class.

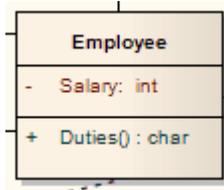
A Class can have *attributes* (data) and *methods* (*operations* or behavior). Classes can inherit characteristics from parent Classes and delegate behavior to other Classes. Class models usually describe the logical structure of the system and are the building blocks from which components are built.

The top section of a Class, as illustrated below, shows the attributes (or data elements) associated with the Class. These hold the 'state' of an object at run-time. If the information is saved to a data store and can be reloaded, it is termed 'persistent'. The lower section contains the Class operations (or methods at run-time). Operations describe the behavior a Class offers to other Classes, and the internal behavior it has (private methods).

Class elements are generally used in Class diagrams and Composite Structure diagrams.

Enterprise Architect also supports a number of stereotyped Class elements to represent various entities in web-page modeling. A Class can also be integrated with an Associate connector to form an Association Class, to allow the Associate connector to have operations and attributes that define certain types of UML

relationship.



Toolbox Icon



See Also:

- [Active Classes](#) ^[943]
- [Parameterized Classes \(Templates\)](#) ^[943]

Learn More:

- [Class Diagrams](#) ^[798]
- [Composite Structure Diagram](#) ^[801]
- [Association Class](#) ^[973]
- [Association](#) ^[970] [on](#) ^[970]
- [Attributes](#) ^[695]
- [Operations](#) ^[707]
- [Web-page Modeling](#) ^[1284] [odeling](#) ^[1284]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, pp. 52-53*) states:

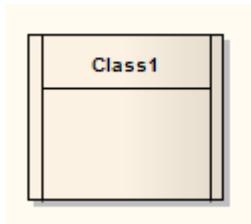
The purpose of a class is to specify a classification of objects and to specify the features that characterize the structure and behavior of those objects.

Objects of a class must contain values for each attribute that is a member of that class, in accordance with the characteristics of the attribute, for example its type and multiplicity.

When an object is instantiated in a class, for every attribute of the class that has a specified default, if an initial value of the attribute is not specified explicitly for the instantiation, then the default value specification is evaluated to set the initial value of the attribute for the object.

Operations of a class can be invoked on an object, given a particular set of substitutions for the parameters of the operation. An operation invocation may cause changes to the values of the attributes of that object. It may also return a value as a result, where a result type for the operation has been defined. Operation invocations may also cause changes in value to the attributes of other objects that can be navigated to, directly or indirectly, from the object on which the operation is invoked, to its output parameters, to objects navigable from its parameters, or to other objects in the scope of the operation's execution. Operation invocations may also cause the creation and deletion of objects.

6.4.2.2.1 Active Classes



Description:

An *Active Class* indicates that, when instantiated, the Class controls its own execution. Rather than being invoked or activated by other objects, it can operate standalone and define its own thread of behavior.

How to:

To define an Active Class in Enterprise Architect, follow the steps below:

Step	Action
1	Highlight a Class, and display its Properties dialog
2	Click on the Advanced button
3	Select the Is Active checkbox
4	Click on the OK button to save the details

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 438*) states:

An active object is an object that, as a direct consequence of its creation, commences to execute its classifier behavior, and does not cease until either the complete behavior is executed or the object is terminated by some external object. (This is sometimes referred to as "the object having its own thread of control.") The points at which an active object responds to communications from other objects is determined solely by the behavior of the active object and not by the invoking object. If the classifier behavior of an active object completes, the object is terminated.

6.4.2.2.2 Parameterized Classes (Templates)

Description:

Enterprise Architect supports *template* or *parameterized Classes*, which specify parameters that must be defined by any binding Class.

Parameterized Classes are commonly implemented in C++; Enterprise Architect imports and generates templated Classes for C++.

A template Class enables its functionality to be reused by any bound Class. If a default value is specified for a parameter, and a binding Class doesn't provide a value for that parameter, the default is used.

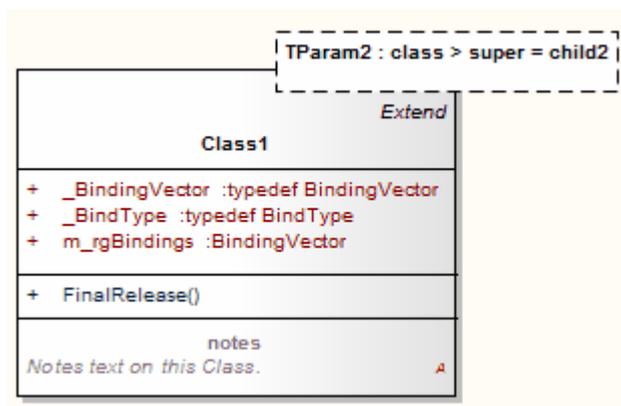
How to:

To create a parameterized Class, follow the steps below:

Step	Action
1	Display the Properties dialog for the required Class
2	Select the Templates page
3	In the Template Parameter(s) panel, click on the Add button The Template Parameter dialog displays
4	Type in the name and type of the parameter and, if required, click on the (...) buttons after the Constraints and Default fields to select the required constraining and default Classes from the Select <Item> dialog The default Class can be either the constraining classifier or any Class that derives from the constraining classifier

Notation Example:

On a diagram, template Classes are shown with the parameters in a dashed outline box in the upper right corner of the Class.

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 622*) states:

A template is a parameterized element that can be used to generate other model elements using TemplateBinding relationships. The template parameters for the template signature specify the formal parameters that will be substituted by actual parameters (or the default) in a binding.

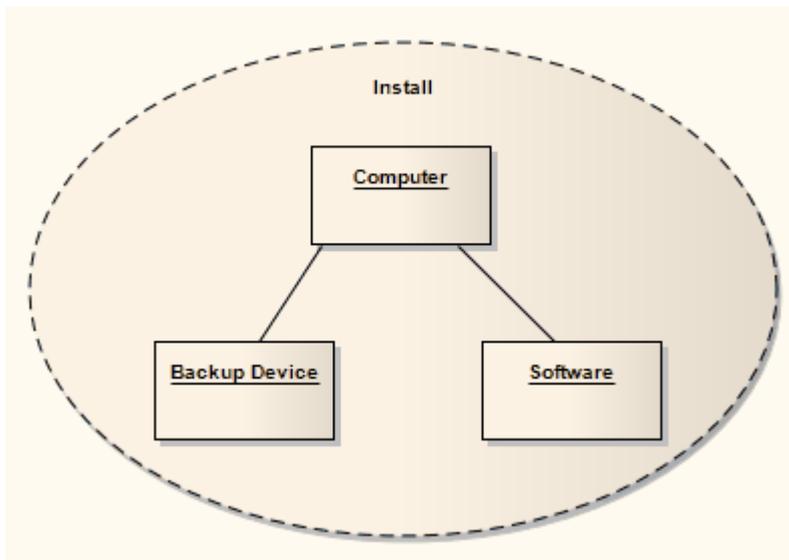
6.4.2.3 Collaboration



Description:

A *Collaboration* defines a set of cooperating roles and their connectors. These are used to collectively illustrate a specific functionality, in a Composite Structure diagram. A Collaboration should specify only the roles and attributes required to accomplish a specific task or function. Although in practice a behavior and its roles could involve many tangential attributes and properties, isolating the primary roles and their requisites simplifies and clarifies the behavior, as well as providing for reuse. A Collaboration often implements a pattern to apply to various situations.

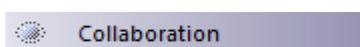
The following example illustrates an *Install* Collaboration, with three roles (Objects) connected as shown. The process for this Collaboration can be demonstrated by attaching an Interaction diagram (Sequence, Timing, Communication or Interaction Overview).



To understand referencing a Collaboration in a specific situation, see the *Collaboration Use* topic.

Enterprise Architect supports a stereotyped Collaboration to represent a Business Use Case Realization in business modeling.

Toolbox Icon:



Learn More:

- [Composite Structure Diagram](#) ^[80]
- [Sequence Diagram](#) ^[84]

- [Timing Diagram](#) ^[830]
- [Communication Diagram](#) ^[859]
- [Interaction Overview Diagram](#) ^[861]
- [Collaboration Use](#) ^[946]
- [Object Element](#) ^[957]
- [Business Use Case Realization](#) ^[1191]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 171*) states:

A collaboration describes a structure of collaborating elements (roles), each performing a specialized function, which collectively accomplish some desired functionality. Its primary purpose is to explain how a system works and, therefore, it typically only incorporates those aspects of reality that are deemed relevant to the explanation.

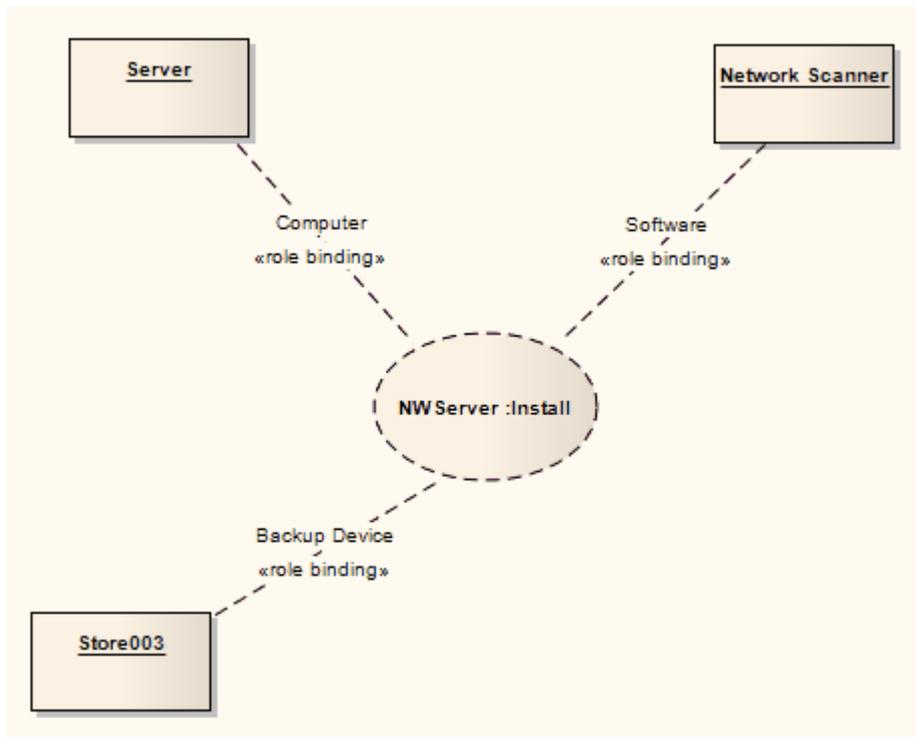
6.4.2.4 Collaboration Use



Description:

Use a *Collaboration Use* to apply a pattern defined by a *Collaboration* to a specific situation, in a *Composite Structure* diagram.

The following example shows a *Use*, *NWServer*, of the *Collaboration Install*, to define the installation process of a network scanner. This process can be defined by an interaction attached to the *Collaboration*. (See the *Collaboration* topic for a representation of the *Install* *Collaboration*.)



To create a Collaboration Use, drag the icon from the Toolbox onto the diagram.

Toolbox Icon:



Learn More:

- [Composite Structure Diagram](#)^[80]
- [Collaboration Element](#)^[945]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 173*) states:

A collaboration use represents one particular use of a collaboration to explain the relationships between the properties of a classifier. A collaboration use shows how the pattern described by a collaboration is applied in a given context, by binding specific entities from that context to the roles of the collaboration. Depending on the context, these entities could be structural features of a classifier, instance specifications, or even roles in some containing collaboration. There may be multiple occurrences of a given collaboration within a classifier, each involving a different set of roles and connectors. A given role or connector may be involved in multiple occurrences of the same or different collaborations.

6.4.2.5 Component



Description

A Component is a modular part of a system, whose behavior is defined by its provided and required interfaces; the internal workings of the Component should be invisible and its usage environment-independent. Source code files, DLLs, Java beans and other artifacts defining the system can be manifested in Components.

A Component can be composed of multiple Classes, or Components pieced together. As smaller Components come together to create bigger Components, the eventual system can be modeled, building-block style, in Component diagrams. By building the system in discrete Components, localization of data and behavior enables decreased dependency between Classes and Objects, providing a more robust and maintainable design.

Toolbox Icon



Learn More:

- [Component Diagram](#)^[80]
- [Class Element](#)^[94]
- [Object Element](#)^[95]

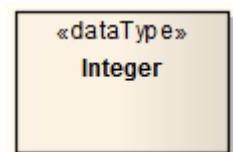
OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 148*) states:

A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment.

A component defines its behavior in terms of provided and required interfaces. As such, a component serves as a type whose conformance is defined by these provided and required interfaces (encompassing both their static as well as dynamic semantics).

6.4.2.6 Data Type



Description

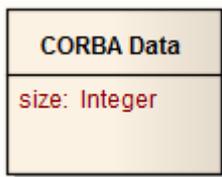
A Data Type is a specific kind of classifier, similar to a Class except that a Data Type cannot own sub Data Types, and instances of a Data Type are identified only by their value. For example, an instance of a Person Class is a Helen object, but an instance of an Integer Data Type is 12.

All copies of an instance of a Data Type, and any instances of that Data Type with the same value, are considered to be the same instance. That is, instances of Helen are not necessarily the same Helen, but all 12s are the same 12. For example, the 12 on a watch face is exactly the same integer as the number of months in a year.

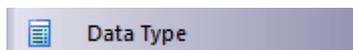
Instances of a Data Type that have attributes (that is, are instances of a structured Data Type) are considered to be the same if the structure is the same and the values of the corresponding attributes are the same. If a Data Type has attributes, instances of that Data Type contain attribute values matching the attributes.

A typical use of Data Types would be to represent programming language primitive types or CORBA basic types. For example, integer and string types are often treated as Data Types.

A Data Type is denoted by a rectangle with the keyword «dataType», as above or, when it is referenced by (for example) an attribute, by a string containing the name of the Data Type, as below:



Toolbox Icon



Learn More:

- [Class Element](#)^[94]

6.4.2.7 Deployment Spec



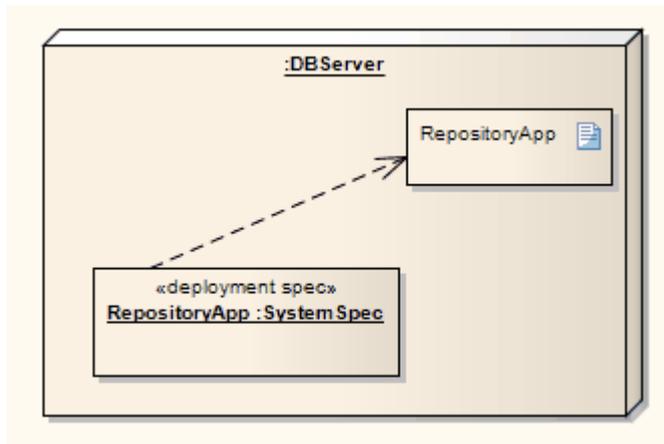
Description

A Deployment Specification (spec) specifies parameters guiding deployment of an artifact, as is necessary with most hardware and software technologies. A specification lists those properties that must be defined for deployment to occur, as represented in a Deployment diagram. An instance of this specification specifies the values for the parameters; a single specification can be instantiated for multiple artifacts.

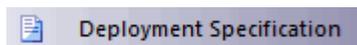
These specifications can be extended by certain component profiles. Examples of standard Tagged Values

that a profile might add to a Deployment Specification are «concurrencyMode» with Tagged Values {thread, process, none} or «transactionMode» with Tagged Values {transaction, nestedTransaction, none}.

The following example depicts the artifact RepositoryApp deployed on the server node, as per the specifications of RepositoryApp, instantiated from the Deployment Specification SystemSpec.



Toolbox Icon



Learn More:

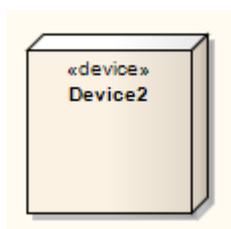
- [Deployment Diagram](#) ^[804]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 206*) states:

A deployment specification specifies a set of properties that determine execution parameters of a component artifact that is deployed on a node. A deployment specification can be aimed at a specific type of container. An artifact that reifies or implements deployment specification properties is a deployment descriptor.

6.4.2.8 Device



Description

A Device is a physical electronic resource with processing capability upon which Artifacts can be deployed for execution, as represented in a Deployment diagram. Complex Devices can consist of other devices; that is, a Device can be a nested element, where a physical machine is decomposed into its elements either

through namespace ownership or through attributes that are typed by Devices.

Toolbox Icon



Learn More:

- [Deployment Diagram](#) ^[804]
- [Artifact Element](#) ^[940]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, 10.3.7, v2.1.1, p. 207*) states:

In the metamodel, a Device is a subclass of Node.

6.4.2.9 Document Artifact



Description

A Document Artifact is an artifact having a stereotype of «document». You create the Document Artifact on a Component, Documentation or Deployment diagram, and associate it with an RTF document. Double-click on the element to display the Linked Document Editor. See the Linked Documents topic. When you have created the linked document, the Document Artifact element on the diagram shows an **A** symbol in the bottom right corner.



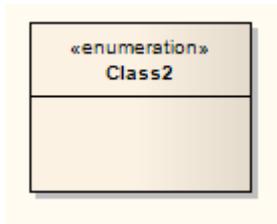
Toolbox Icon



Learn More:

- [Linked Documents](#) ^[729]
- [Artifact Element](#) ^[940]

6.4.2.10 Enumeration



Description

An *Enumeration* is a data type, whose instances can be any of a number of user-defined enumeration literals. It is possible to extend the set of applicable enumeration literals in other packages or profiles. You create Enumerations in Class or Package diagrams, and in diagrams developed from the Metamodel and Profile pages of the **Toolbox**.

Toolbox Icon



Learn More:

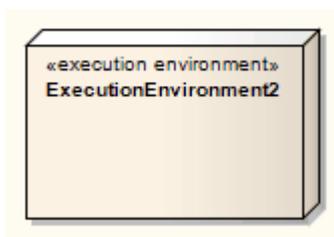
- [Class Diagram](#) ^[798]
- [Package Diagram](#) ^[796]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 69*) states:

An enumeration is a data type whose values are enumerated in the model as enumeration literals.

6.4.2.11 Execution Environment

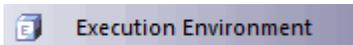


Description

An Execution Environment is a node that offers an execution environment for specific types of components that are deployed on it in the form of executable artifacts. This is depicted in a Deployment diagram.

Execution Environments can be nested; for example, a database Execution Environment can be nested in an operating system Execution Environment. Components of the appropriate type are then deployed to specific Execution Environment nodes.

Toolbox Icon



Learn More:

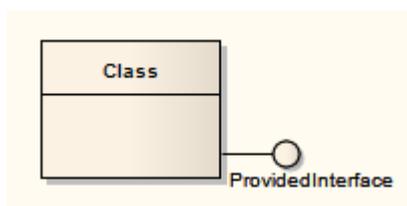
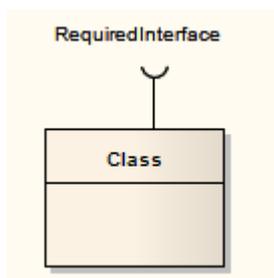
- [Deployment Diagram](#)^[804]
- [Node Element](#)^[956]
- [Component Element](#)^[948]
- [Artifact Element](#)^[940]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 210*) states:

... an ExecutionEnvironment is ... usually part of a general Node, representing the physical hardware environment on which the ExecutionEnvironment resides. In that environment, the ExecutionEnvironment implements a standard set of services that Components require at execution time (at the modeling level these services are usually implicit). For each component Deployment, aspects of these services may be determined by properties in a DeploymentSpecification for a particular kind of ExecutionEnvironment.

6.4.2.12 Expose Interface



Description

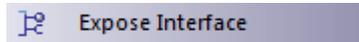
The Expose Interface element is a graphical method of depicting the required or supplied interfaces of a Component, Class or Part, in a Component or Composite Structure diagram. It just identifies the fact that the element provides or requires an interface; to depict the fact that the provided interface is used, or the required interface provided, by another element use the Assembly connector.

The Expose Interface element must be attached to the Class or Component element, and it becomes a child element of that Class or Component; it cannot exist independently. You can attach more than one Expose Element to another element.

When you create the Expose Interface element, a dialog displays in which you enter a name for the element

and specify whether it represents a required interface or a provided interface.

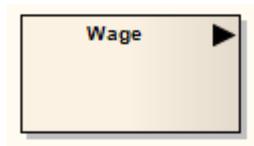
Toolbox Icon



Learn More:

- [Component Diagram](#)^[807]
- [Composite Structure Diagram](#)^[807]
- [Interface](#)^[955]
- [Component](#)^[946]
- [Class](#)^[947]
- [Part](#)^[960]
- [Assembly](#)^[969]

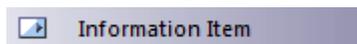
6.4.2.13 Information Item



Description

An Information Item represents an abstraction of data. It is used in Activity, Analysis and Object diagrams. An Information Item is also represented by an Information Flow connector.

Toolbox Icon



Learn More:

- [Information Flow](#)^[983]
- [Activity Diagram](#)^[817]
- [Analysis Diagram](#)^[1187]
- [Object Diagram](#)^[799]

OMG UML Specification

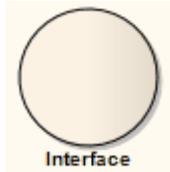
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 608*) states:

An information item is an abstraction of all kinds of information that can be exchanged between objects. It is a kind of classifier intended for representing information at a very abstract way, one which cannot be instantiated.

One purpose of information items is to be able to define preliminary models, before having made detailed modeling decisions on types or structures. One other purpose of information items and information flows is to abstract complex models by a less precise but more general representation of the information exchanged

between entities of a system.

6.4.2.14 Interface

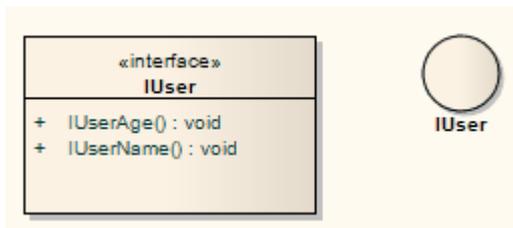


Description

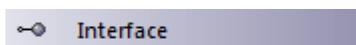
An *Interface* is a specification of behavior (or contract) that implementers agree to meet. By implementing an Interface, *Classes* are guaranteed to support a required behavior, which enables the system to treat non-related elements in the same way; that is, through the common interface. You also use Interfaces in a Composite Structure diagram.

Interfaces are drawn in a similar way to a Class, with operations specified, as shown below. They can also be drawn as a circle with no explicit operations detailed. Right-click on the element and select the **Use Circle Notation** context menu option to switch between styles. Realize connectors to an Interface drawn as a circle are drawn as a solid line without target arrows.

An Interface cannot be instantiated (that is, you cannot create an object from an Interface). You must create a Class that 'implements' the Interface specification, and in the Class body place operations for each of the Interface operations. You can then instantiate the Class.



Toolbox Icon



Learn More:

- [Class Diagram](#) ^[798]
- [Composite Structure Diagram](#) ^[801]
- [Realization](#) ^[1007]

OMG UML Specification

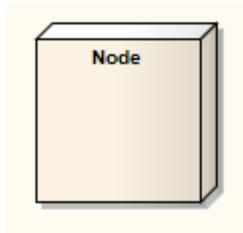
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 88*) states:

An interface is a kind of classifier that represents a declaration of a set of coherent public features and obligations. An interface specifies a contract; any instance of a classifier that realizes the interface must fulfill that contract. The obligations that may be associated with an interface are in the form of various kinds of constraints (such as pre- and post-conditions) or protocol specifications, which may impose ordering

restrictions on interactions through the interface.

Since interfaces are declarations, they are not instantiable. Instead, an interface specification is implemented by an instance of an instantiable classifier, which means that the instantiable classifier presents a public facade that conforms to the interface specification. Note that a given classifier may implement more than one interface and that an interface may be implemented by a number of different classifiers.

6.4.2.15 Node



Description

A Node is a physical piece of equipment on which the system is deployed, such as a workgroup server or workstation. A Node usually hosts components and other executable pieces of code, which again can be connected to particular processes or execution spaces. Typical Nodes are client workstations, application servers, mainframes, routers and terminal servers.

Nodes are used in Deployment diagrams to model the deployment of a system, and to illustrate the physical allocation of implemented artifacts. They are also used in web modeling, from dedicated web modeling pages in the Toolbox.

Toolbox Icon



Learn More:

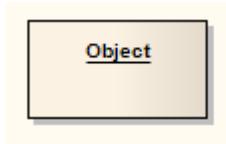
- [Deployment Diagram](#) ^[804]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 213*) states:

In the metamodel, a Node is a subclass of Class. It is associated with a Deployment of an Artifact. It is also associated with a set of Elements that are deployed on it. This is a derived association in that these PackageableElements are involved in a Manifestation of an Artifact that is deployed on the Node. Nodes may have an internal structure defined in terms of parts and connectors associated with them for advanced modeling applications.

6.4.2.16 Object

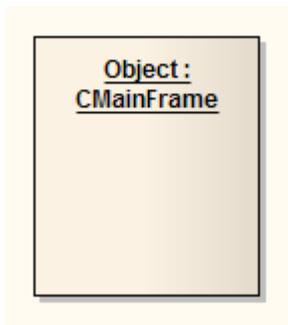


Description

An Object is a particular instance of a Class at run time. For example a car with the license plate AAA-001 is an instance of the general class of cars with a license plate number attribute. Objects are often used in analysis to represent the numerous artifacts and items that exist in any business, such as pieces of paper, faxes and information. To model the varying behavior of Objects at run-time, use run-time states.

Early in analysis, Objects can be used to quickly capture all the things that are of relevance within the system domain, in an Object, Composite Structure or Communication diagram. As the model progresses these analysis Objects are refined into generic Classes from which instances can be derived to represent common business items. Once Classes are defined, Objects can be typed; that is they can have a classifier set that indicates their base type. See the Object Classifiers topic.

Enterprise Architect also supports a number of stereotyped Object elements to represent various entities in business modeling.



Toolbox Icon



See Also:

- [Run-time State](#) ^[958]
- [Define a Run-time Variable](#) ^[958]
- [Remove a Defined Variable](#) ^[959]
- [Object State](#) ^[959]

Learn More:

- [Object Diagram](#) ^[799]
- [Composite Structure Diagram](#) ^[801]
- [Communication Diagram](#) ^[859]
- [Class](#) ^[941]

- [Object Classifiers](#)^[705]
- [Business Interaction Objects](#)^[1197]

6.4.2.16.1 Run-time State

At run-time, an Object instance can have specific values for its attributes, or exist in a particular state. To model the varying behavior of Objects at run-time, use instance values selected from the Select <Item> dialog and run-time states or run-states.

Typically there is interest in the run-time behavior of Objects that already have a classifier set. You can select from the classifier's attribute list and apply specific values for your Object instance. If the classifier has a child State Machine, its States propagate to a list where the run-time state for the Object can be defined.

Example

The following example defines run-time values for the listed variables, which are attributes of the instances' classifier *AccountItem*.



See Also:

- [Define a Run-Time Variable](#)^[958]
- [Remove a Defined Variable](#)^[959]

Learn More:

- [Object](#)^[957]
- [Object State](#)^[959]

6.4.2.16.1.1 Define a Run-time Variable

How to:

To add run-time state instance variables to an Object, follow the steps below:

Step	Action
1	Right-click on the Object The context menu displays
2	If Instance Variables are supported, select the Advanced Set Run State menu option (or press (Ctrl+Shift+R)) The Set Run State dialog displays
3	In the Variable field, click on the drop-down arrow and select the variable, or type in the new variable name

Step	Action
4	Set the Operator , the Value and optionally type in a Note
5	Click on the OK button to save the variable

6.4.2.16.1.2 Remove a Defined Variable

How to:

To delete a run-time state variable for an Object, follow the steps below:

Step	Action
1	Right-click on the required Object The context menu displays
2	Select the Set Run State option The Run State dialog displays
3	In the Variable field, click on the drop-down arrow and select the variable to delete
4	Clear the Value field
5	Click on the OK button

6.4.2.16.2 Object State

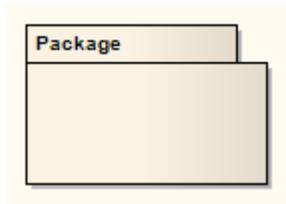
How to:

To set the Object state for a Class instance, follow the steps below:

Step	Action
1	Right-click on the required Object and select the Advanced Set Object State context menu option The Set Instance State dialog displays
2	In the State field, either type the required State (such as Awaiting Approval) or select a State from the drop-down list The drop-down list for the State field is populated with: <ol style="list-style-type: none"> 1. Any States owned by the object's classifier 2. Any States owned by any superclasses of the object's classifier 3. Any States owned by State Machines owned by the object's classifier 4. Any States owned by State Machines owned by any superclasses of the object's classifier
3	Click on the OK button to apply the State The object now shows the run-time state in square brackets below the object name

Step	Action
	

6.4.2.17 Package



Description

A *Package* is a namespace as well as an element that can be contained in other Package's namespaces. A Package can own or merge with other Packages, and its elements can be imported into a Package's namespace. In addition to using Packages in the **Project Browser** to organize your project contents, you can drag these Packages onto a diagram workspace (most diagram types, both standard and extended) for structural or relational depictions, including Package imports or merges.

Toolbox Icon



OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 109*) states:

A package is a namespace for its members, and may contain other packages. Only packageable elements can be owned members of a package. By virtue of being a namespace, a package can import either individual members of other packages, or all the members of other packages. In addition a package can be merged with other packages.

6.4.2.18 Part



Description

Parts are run-time instances of Classes or Interfaces. Multiplicity can be specified for a Part, using the notation:

(x{...y})

where x specifies the initial or set amount of instances when the composite structure is created, and y indicates the maximum amount of instances at any time.

Parts are used to express composite structures, or modeling patterns that can be invoked by various objects to accomplish a specific purpose. When illustrating the composition of structures, Parts can be embedded as properties of other Parts. When embedded as properties, Parts can be bordered by a solid outline, indicating the surrounding Part owns the Part by composition. Alternatively, a dashed outline indicates that the property is referenced and used by the surrounding Part, but is not composed within it.

Toolbox Icon



Learn More:

- [Composite Structure Diagram](#) [80†]
- [Class](#) [94†]
- [Interface](#) [95†]
- [Properties](#) [803†]
- [Add Property Value](#) [96†]

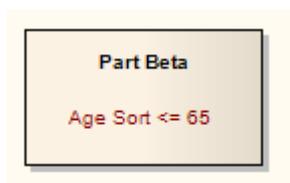
6.4.2.18.1 Add Property Value

How to:

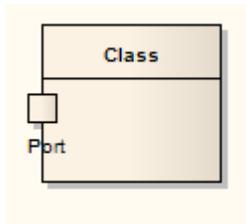
To add property value variables to a Part, follow the steps below:

Step	Action
1	Right-click on the Part The context menu displays
2	Select the Advanced Set Property Values menu option (or press (Ctrl+Shift+R)) The Set Property Values dialog displays
3	In the Variable field, click on the drop-down arrow and select the variable, or type in the new variable name
4	Set the Operator , the Value and optionally type in a Note
5	Click on the OK button to save the variable

A Part with a property value resembles the following figure.



6.4.2.19 Port



Description

Ports define the interaction between a classifier and its environment. Interfaces controlling this interaction can be depicted using the Interface element. Any connector to a Port must provide the required interface, if defined. Ports can appear on a contained Part, a Class, or the boundary of a Composite element.

A Port is a *typed* structural feature or property of its containing classifier. Ports are typically created in Class diagrams, Object diagrams and Composite Structure diagrams.

Toolbox Icon



Learn More:

- [Add a Port to an Element](#)^[962]
- [Inherited and Redefined Ports](#)^[963]
- [The Property Tab](#)^[963]
- [Composite Structure Diagram](#)^[801]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 182*) states:

A port is a property of a classifier that specifies a distinct interaction point between that classifier and its environment or between the (behavior of the) classifier and its internal parts. Ports are connected to properties of the classifier by connectors through which requests can be made to invoke the behavioral features of a classifier. A Port may specify the services a classifier provides (offers) to its environment as well as the services that a classifier expects (requires) of its environment.

6.4.2.19.1 Add a Port to an Element

How to:

To add a new Port to an element, use one of the following steps:

Step	Action
1	Click on the Port symbol in the Composite Elements page of the Toolbox, and drag the symbol to (or click on) the target host element This creates an untyped, simple Port on the boundary, near the cursor position

Step	Action
2	On the context menu of a suitable Class, Part or Composite element, select the Embedded Elements Add Port menu option to add a new Port at the cursor position
3	Drag a suitable classifier from the Project Browser onto a Class or Part Enterprise Architect prompts you to add a typed Port or Part at the cursor position The new Port is typed by the original dragged classifier
4	Use the Embedded Elements window to add a new Port to the currently selected element

Learn More:

- [Composite Elements](#)^[647]
- [Inherited and Redefined Ports](#)^[963]

6.4.2.19.2 *Inherited and Redefined Ports*

A Port is a *redefinable* and *re-useable* property of a composite classifier such as a Component. A Component can inherit Ports from its parent; if a Component's parent owns Ports, when you open the Embedded Elements dialog for the Component and select the **Show Owned/Inherited** checkbox the inherited Ports and their named owners are listed.

If you want to show an inherited Port on a Component, the Embedded Elements dialog provides two options:

- Expose an inherited Port - tick the checkbox next to the Port's name, to create a read-only copy of the Port; this is convenient for modeling Port interactions in child elements where the Ports are defined in the parent elements
- Redefine an inherited Port - select the row for the Port and click on the **Redefine** button, to create an editable copy of the Port; this is useful where a child element places additional restrictions or behavior on the Port

The above is also true for Components that inherit Ports from realized Interfaces, and for Component instances that inherit Ports from their classifying Component.

6.4.2.19.3 *The Property Page*

The element Properties dialog for Ports and Parts has a Property page in place of the Class element Details page.

This page defines the type, initial value, Qualifiers, multiplicity, and redefined and subsetted properties of the Port or Part.

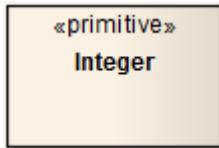
You set the Qualifiers by clicking on the **Qualifiers** button, to display the Qualifiers dialog.

You add **Redefined** and **Subsetted Properties** by clicking on the appropriate **Add** button, to display the Select Property dialog.

Learn More:

- [Qualifiers Dialog](#)^[972]
- [Select Property Dialog](#)^[692]
- [Inherited and Redefined Ports](#)^[963]

6.4.2.20 Primitive



Description:

A Primitive element identifies a predefined data type, without any relevant substructure (that is, it has no parts in the context of UML). It could be regarded as a conceptual Data Type.

Toolbox Icon:



Learn More:

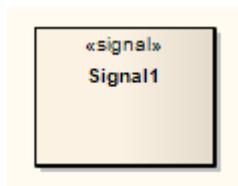
- [Data Type Element](#)^[948]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 124*) states:

A primitive data type may have an algebra and operations defined outside of UML, for example, mathematically ... The run-time instances of a primitive type are data values. The values are in many-to-one correspondence to mathematical elements defined outside of UML (for example, the various integers). Instances of primitive types do not have identity. If two instances have the same representation, then they are indistinguishable.

6.4.2.21 Signal



Description:

A Signal is a specification of Send request instances communicated between objects, typically in a Class or Package diagram. The receiving object handles the Received request instances as specified by its receptions. The data carried by a Send request is represented as attributes of the Signal. A Signal is defined independently of the classifiers handling the signal occurrence.

To define a reception, create an operation in the receiving object and assign the stereotype <<signal>> to it. The reception has the same name as the signal that the object can receive.

Toolbox Icon:**Learn More:**

- [Class Diagram](#)^[798]
- [Package Diagram](#)^[796]
- [Send Element](#)^[917]
- [Receive Element](#)^[915]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 450*) states:

A signal triggers a reaction in the receiver in an asynchronous way and without a reply. The sender of a signal will not block waiting for a reply but continue execution immediately. By declaring a reception associated to a given signal, a classifier specifies that its instances will be able to receive that signal, or a subtype thereof, and will respond to it with the designated behavior.

And (*UML Superstructure Specification, v2.1.1, p. 447 - 448*):

A reception is a declaration stating that a classifier is prepared to react to the receipt of a signal. A reception designates a signal and specifies the expected behavioral response. The details of handling a signal are specified by the behavior associated with the reception or the classifier itself. ...Receptions are shown using the same notation as for operations with the keyword <signal>

6.5 UML Connectors

A *connector* is a logical or functional relationship between model elements. There are several different connector types, each having a particular purpose and syntax. Enterprise Architect supports all of the UML connectors as well as various custom connectors. Together with the UML Elements, these form the basis of UML models.

For more information on using these connectors, consult the appropriate topic by clicking on the required connector icon in the table below.

Behavioral Diagram Connectors	Structural Diagram Connectors	Inbuilt and Extended Connectors
Activity Diagrams <ul style="list-style-type: none">  Control Flow  Object Flow  Interrupt Flow 	Composite Structure Diagrams <ul style="list-style-type: none">  Connector  Assembly  Delegate  Role Binding  Represents  Occurrence 	Analysis Diagrams <ul style="list-style-type: none">  Information Flow  Object Flow  Associate  Realize  Representation
Use Case Diagrams <ul style="list-style-type: none">  Use  Associate  Generalize  Include  Extend  Realize  Invokes  Precedes 	Package and Class Diagrams <ul style="list-style-type: none">  Associate  Generalize  Compose  Aggregate  Association Class  Assembly  Realize  Nesting  Package Merge  Package Import 	Common Connectors <ul style="list-style-type: none">  Dependency  Realize  Trace  Information Flow  Note Link
State Diagrams <ul style="list-style-type: none">  Transition  Object Flow 		Profile <ul style="list-style-type: none">  Extension  Generalize  Application  Tagged Value  Redefinition
Timing Diagrams		

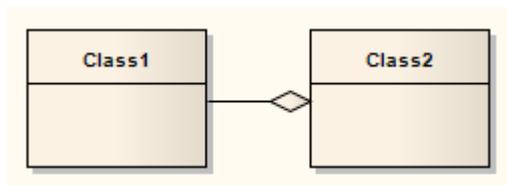
Behavioral Diagram Connectors	Structural Diagram Connectors	Inbuilt and Extended Connectors
<p>⇒ Message</p>	<p>Component Diagrams</p> <p>⊖ Assembly</p> <p>↳ Delegate</p> <p>↗ Associate</p> <p>↗ Realize</p> <p>↗ Generalize</p>	<p>Metamodel</p> <p>↗ Generalize</p> <p>↗ Associate</p> <p>↗ Compose</p> <p>↗ Aggregate</p>
<p>Sequence Diagrams</p> <p>⇒ Message</p> <p>↻ Self-Message</p> <p>⏮ Recursion</p> <p>⏮ Call</p>	<p>Deployment Diagrams</p> <p>↗ Associate</p> <p>↗ Communication Path</p> <p>↗ Association Class</p> <p>↗ Generalize</p> <p>↗ Realize</p> <p>↗ Deployment</p> <p>↗ Manifest</p> <p>↗ Nesting</p>	<p>Custom</p> <p>↗ Associate</p> <p>↗ Aggregate</p> <p>↗ Generalize</p> <p>↗ Realize</p> <p>↗ Nesting</p>
<p>Communication Diagrams</p> <p>↗ Associate</p> <p>↗ Realize</p> <p>↗ Nesting</p>	<p>User Interface</p> <p>↗ Associate</p> <p>↗ Aggregate</p> <p>↗ Generalize</p> <p>↗ Realize</p>	<p>Requirements</p> <p>↗ Aggregate</p> <p>↗ Inheritance</p> <p>↗ Associate</p> <p>↗ Implements</p>
<p>Interaction Overview Diagrams</p> <p>↗ Control Flow</p> <p>↗ Object Flow</p> <p>↗ Interrupt Flow</p>	<p>Maintenance</p> <p>↗ Associate</p> <p>↗ Aggregate</p>	<p>WSDL</p> <p>No special connectors</p>
<p>XML Schema</p> <p>↗ Generalize</p> <p>↗ Associate</p>	<p>Object</p> <p>↗ Information Flow</p>	<p>Documentation</p> <p>No special connectors</p>

Behavioral Diagram Connectors	Structural Diagram Connectors	Inbuilt and Extended Connectors
Data Modeling	 Associate	
 Associate	 Dependency	

Notes:

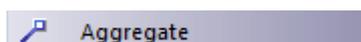
- Invokes and Precedes relationships are defined by the Open Modeling Language (OML). They are stereotyped Dependency relationships; Invokes indicates that Use Case A, at some point, causes Use Case B to happen, whilst Precedes indicates that Use Case C must complete before Use Case D can begin.
- An Extension relationship shows that a stereotype extends one or more metaclasses. All stereotypes must extend either one or more Metaclasses, or another stereotype that extends a stereotype (that itself extends a stereotype, and so on).
- A Tagged Value relationship defines a reference-type (that is, RefGUID) Tagged Value owned by the source stereotype. The Tagged Value is named for the target role of this association, and is limited to referencing elements with the stereotype by the association target element.
- The Application and Redefinition relationships are **deprecated**.

6.5.1 Aggregation

**Description:**

An Aggregation connector is a type of association that shows that an element contains or is composed of other elements. It is used in Class models, Package models and Object models to show how more complex elements (aggregates) are built from a collection of simpler elements (component parts; for example, a car from wheels, tires, motor and so on).

A stronger form of aggregation, known as Composite Aggregation, is used to indicate ownership of the whole over its parts. The part can belong to only one Composite Aggregation at a time. If the composite is deleted, all of its parts are deleted with it.

Toolbox Icon:**Learn More**

- [Class Diagram](#) ^[798]
- [Package Diagram](#) ^[796]
- [Object Diagram](#) ^[799]
- [Change Aggregation Connector Form](#) ^[969]

6.5.1.1 Change Aggregation Connector Form

In Enterprise Architect, the default Aggregation relationship is the weak form of the relationship, represented by a hollow diamond.

To change the form of an Aggregation connector from weak to strong, follow the steps below:

Step	Action
1	Right-click on an Aggregation connector to display the context menu
2	Select Set Aggregation to Composite The diamond is shown as filled

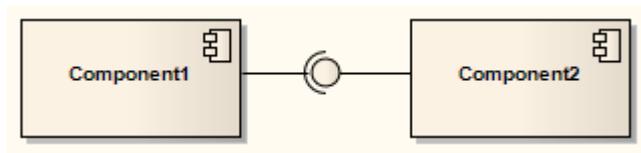
Notes:

- If the connector is already a Strong (Composition) connector, the context menu option changes to **Set Aggregation to Shared**

Learn More:

- [Aggregation](#) ^[968]

6.5.2 Assembly



Description:

An Assembly connector bridges a component's required interface (Component1) with the provided interface of another component (Component2), typically in a Component diagram.

Toolbox Icon



Learn More:

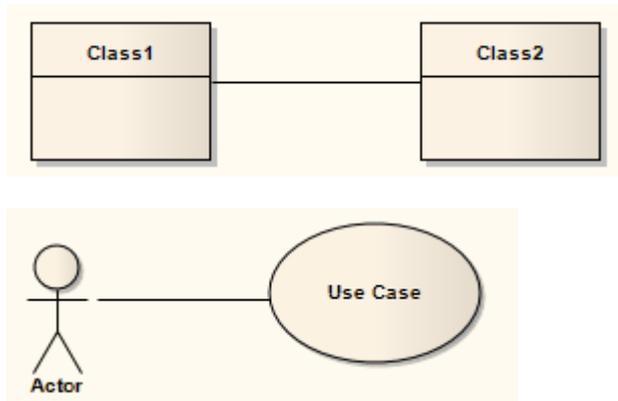
- [Component Diagram](#) ^[807]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 156*) states:

An assembly connector is a connector between two components that defines that one component provides the services that another component requires. An assembly connector is a connector that is defined from a required interface or port to a provided interface or port.

6.5.3 Association



Description:

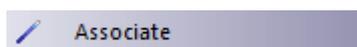
An Association implies two model elements have a relationship, usually implemented as an instance variable in one Class. This connector can include named roles at each end, multiplicity, direction and constraints. Association is the general relationship type between elements. To connect more than two elements in an association, you can use the N-Ary Association element.

When code is generated for Class diagrams, Associations become instance variables in the target Class. The relationship is also used in Package, Object, Communication, Data Modeling and Deployment diagrams.

An Associate connector can also be integrated with a Class element to form an Association Class, to allow the Associate connector to have operations and attributes that define certain types of UML relationship.

You can also define template binding parameters for an Associate connector between a binding Class and a parameterized Class.

Toolbox Icon:



Learn More

- [N-Ary Association](#) ^[1296]
- [Association Class](#) ^[973]
- [Class Diagrams](#) ^[798]
- [Template Binding](#) ^[1010]
- [Qualifiers](#) ^[971]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 41*) states:

An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type.

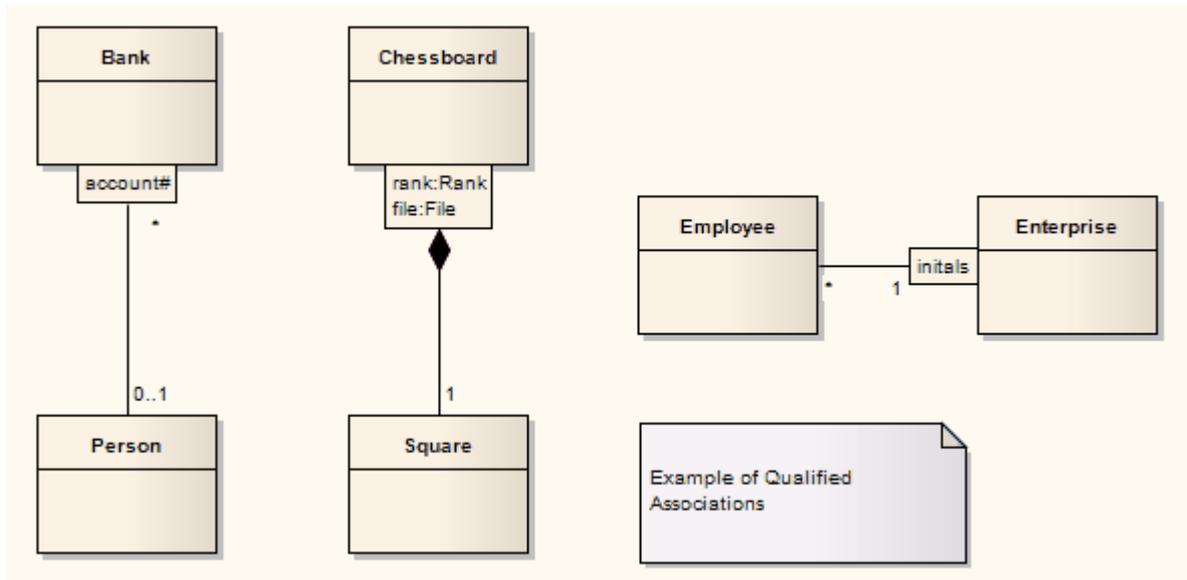
An end property of an association that is owned by an end class or that is a navigable owned end of the association indicates that the association is navigable from the opposite ends; otherwise, the association is

not navigable from the opposite ends.

6.5.3.1 Qualifiers

Qualifiers are ordered sets of properties of an Association end point, a Part, a Port, or an Attribute, that limit the nature of the relationship between two classifiers or objects. You define a qualifier on the **Qualifiers** dialog, which you display by clicking on the (...) button at the end of the **Qualifiers** field on the Association, Part, Port or Attribute **Properties** dialog.

Examples



Notes:

- When typing multiple Qualifiers into the **Qualifier(s)** field on a **Properties** dialog, separate them with a semi-colon; each Qualifier then displays on a separate line. For example, in the diagram the Qualifier 'rank:Rank;file:File' has been rendered in two lines, with a line break at the ; character.
- You can enable or disable Qualifier rectangles in the **Diagram** page of the **Options** dialog (select the **Tools | Options | Diagram** menu option). If disabled, the old style text Qualifiers are used. It is not recommended that you disable Qualifiers as they are an integral part of the UML.
- You can enable or disable a mild shading on the Qualifier rectangles in the **Links** page of the **Options** dialog.

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 129*) states:

A qualifier declares a partition of the set of associated instances with respect to an instance at the qualified end (the qualified instance is at the end to which the qualifier is attached). A qualifier instance comprises one value for each qualifier attribute. Given a qualified object and a qualifier instance, the number of objects at the other end of the association is constrained by the declared multiplicity. In the common case in which the multiplicity is 0..1, the qualifier value is unique with respect to the qualified object, and designates at most one associated object. In the general case of multiplicity 0.., the set of associated instances is partitioned into subsets, each selected by a given qualifier instance. In the case of multiplicity 1 or 0..1, the qualifier has both semantic and implementation consequences. In the case of multiplicity 0..*, it has no real semantic consequences but suggests an implementation that facilitates easy access of sets of associated instances linked by a given qualifier value.*

6.5.3.1.1 Qualifiers Dialog

The **Qualifiers** dialog is used to define the [Qualifiers](#) [971] of an [Association connector end](#) [758], [Port](#) [962], [Part](#) [960] or [Attribute](#) [695].

General Tab

Review, edit or complete the fields as indicated in the following table.

Field	Action	See also
Name	Display the name of the Qualifier. For a new Qualifier, type the name (with no spaces).	
Alias	Display an optional alias for the Qualifier. If necessary, type in a new alias.	
Type	<p>Display the Qualifier type.</p> <p>The type can be defined by the code language (data type) or by a classifier element. When you click on the drop-down arrow, the set of values in the list provides the appropriate data types.</p> <p>To select or define possible classifiers, either click on the Select Type option in the list, or click on the (...) (Select) button to display the Select <Item> dialog.</p> <p>To add new code language data types that can be displayed in this list, see the Data Types topic.</p>	Select <Item> [690] dialog Data Types [771]
Scope	Define the Qualifier as Public , Protected , Private or Package . If necessary, click on the drop-down arrow and select a different scope.	
Stereotype	Define the optional stereotype of the Qualifier. If necessary, either type a different stereotype name or click on the drop-down arrow and select a stereotype.	
Derived	Indicate that the Qualifier is a calculated value. If you select this checkbox, the Qualifier name on the element has the derived symbol (<i>f</i>) as a prefix.	
Static	Indicate that the Qualifier is a static member.	
Const	Indicate that the Qualifier is a constant.	
Initial	Display an optional initial value. If necessary, type in a new initial value.	
Notes	Enter any free text notes associated with the Qualifier. You can format the notes text using the Notes toolbar at the top of the field.	Notes [770] toolbar

To change the position of a Qualifier in the list in the **Qualifiers** panel, click on the **Scroll Up** or **Scroll Down** (hand) buttons.

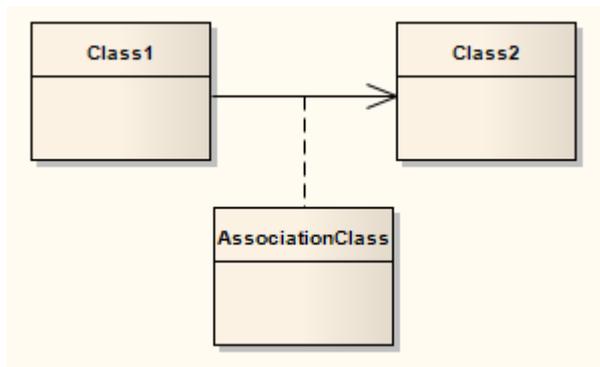
Detail Tab

Use the **Detail** tab to model additional properties of a selected Qualifier, such as its multiplicity, redefined properties and subsetted properties.

Select a Qualifier on the **General** tab, then review, edit or complete the **Detail** tab fields as indicated in the following table.

Field	Action	See also
Multiplicity		
Lower bound	Define a lower limit to the number of elements allowed in the collection.	
Upper bound	Define an upper limit to the number of elements allowed in the collection.	
Allow Duplicates	Indicate that duplicates are allowed. Maps to the UML property <i>isUnique</i> , value <i>FALSE</i>).	
Multiplicity is Ordered	Indicate that the collection is ordered.	
Redefined Property	Review the redefined properties for the Qualifier. Add redefined properties by clicking on the Add button to display the Select Property dialog.	Select Property ^[692] dialog
Subsetted Property	Review the subsetted properties for the qualifier. Add subsetted properties by clicking on the Add button to display the Select Property dialog.	Select Property ^[692] dialog

6.5.4 Association Class



Description

An *Association Class* is a UML construct that enables an Association to have *attributes* and *operations* (*features*). This results in a hybrid relation with the characteristics of an Association and a Class.

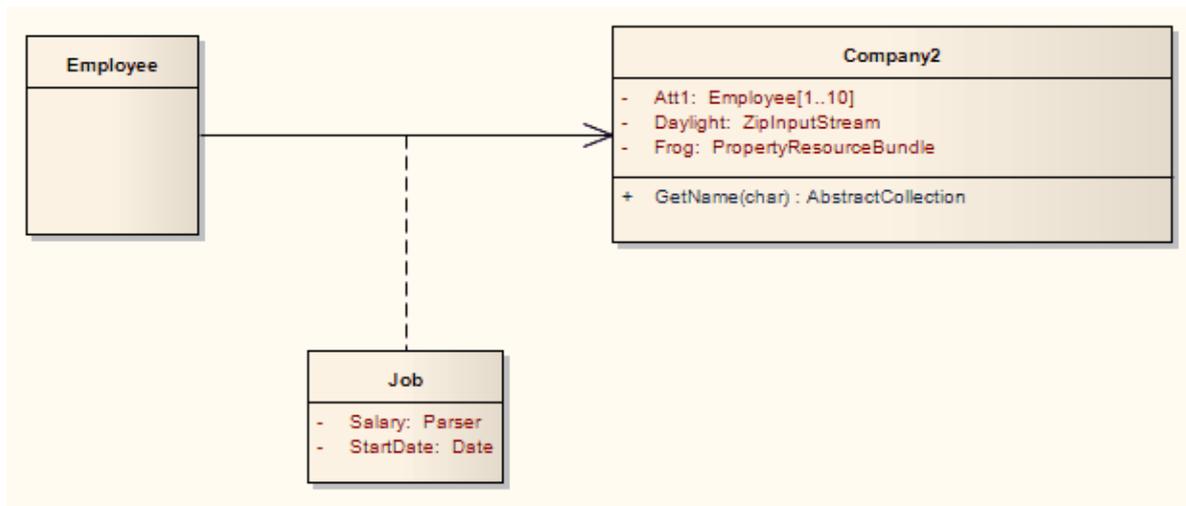
When you add an Association Class connection, Enterprise Architect also creates a Class that is automatically connected to the Association. When you hide or delete the Association, the Class is also hidden or deleted.

To add an Association Class to a Class or Deployment diagram, click on the *Association Class* icon in the **Toolbox**. Click and hold on the source object in the diagram while you drag the line to the target element, then release the mouse button. Enterprise Architect draws the connector and adds the Class, then prompts you to add the Class name. Note that the names of the Class and the connector are the same. You can also connect a new Class to an existing Association.

You can highlight the Class part of an Association Class in the **Project Browser**, by selecting the **Find Association Class** context menu option on the Association connector.

Example

The following diagram illustrates an Association Class between model elements. Note the dotted line from the Class to the Association. You cannot move or delete this line.



Notes:

- If you are applying a stereotype with a Shape Script to an Association Class, be aware that the Shape Script is applied to both the Class part and the Association part. Therefore, you might have to include logic in the shape main that tests the type of the element so that you can give separate drawing instructions for Class and for Association. Such logic is not necessary in the:
 - shape source or shape target, which are ignored by Classes, or the
 - decoration shapes, which are ignored by Association connectors.
 If you dissociate the Class from the Association connector, both parts keep their Shape Scripts until the stereotypes are removed.

Toolbox Icon



Learn More:

- [Class Diagram](#) ^[798]
- [Connect New Class to Association](#) ^[975]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 49*) states:

A model element that has both association and class properties. An AssociationClass can be seen as an association that also has class properties, or as a class that also has association properties. It not only connects a set of classifiers but also defines a set of features that belong to the relationship itself and not to

any of the classifiers.

6.5.4.1 Connect New Class to Association

How to:

To connect a new Class to an existing Association, follow the steps below:

Step	Action
1	Create a Class in the diagram containing the Association to connect
2	Right-click on the new Class The context menu displays
3	Select the Advanced Association Class menu option The Create Association Class dialog displays
4	Select the connector to connect to
5	Click on the OK button

Learn More:

- [Association Class](#)^[973]

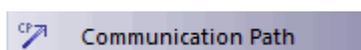
6.5.5 Communication Path



Description

A Communication Path defines the path through which two DeploymentTargets are able to exchange signals and messages. Communication Path is a specialization of Association. A DeploymentTarget is the target for a deployed Artifact and can be a Node, Property or InstanceSpecification in a Deployment diagram.

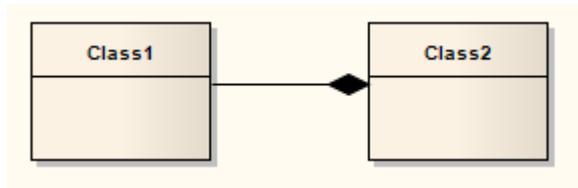
Toolbox Icon



Learn More:

- [Deployment Diagram](#)^[804]
- [Association](#)^[970]

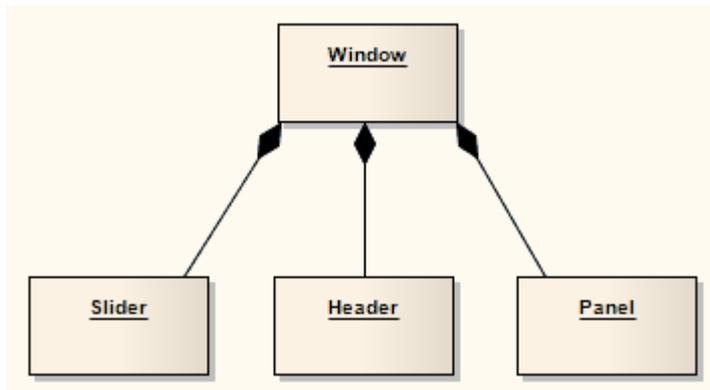
6.5.6 Composition



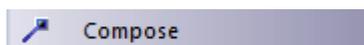
Direction

A Composition is used to depict an element that is made up of smaller components, typically in a Class or Package diagram. A component - or part instance - can be included in a maximum of one composition at a time. If a composition is deleted, usually all of its parts are deleted with it; however, a part can be individually removed from a composition without having to delete the entire composition. Compositions are transitive, asymmetric relationships and can be recursive.

Example



Toolbox Icon



Learn More

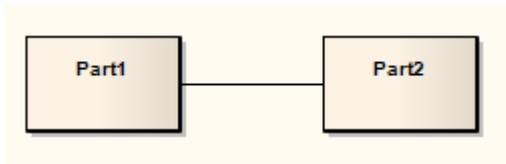
- [Aggregation](#) ^[968]
- [Class Diagram](#) ^[798]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 43*) states:

Composite aggregation is a strong form of aggregation that requires a part instance be included in at most one composite at a time. If a composite is deleted, all of its parts are normally deleted with it.

6.5.7 Connector



Description

Connectors illustrate communication links between parts to fulfill the structure's purpose, typically in a Composite Structure diagram. Each Connector end is distinct, controlling the communication pertaining to its connecting element. These elements can define constraints specifying this behavior. Connectors can have multiplicity.

Toolbox Icon



Learn More

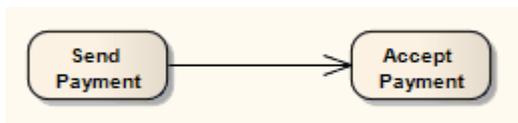
- [Composite Structure Diagram](#) 

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 177*) states:

Specifies a link that enables communication between two or more instances. This link may be an instance of an association, or it may represent the possibility of the instances being able to communicate because their identities are known by virtue of being passed in as parameters, held in variables or slots, or because the communicating instances are the same instance. The link may be realized by something as simple as a pointer or by something as complex as a network connection. In contrast to associations, which specify links between any instance of the associated classifiers, connectors specify links between instances playing the connected parts only.

6.5.8 Control Flow



Description

The *Control Flow* is a connector connecting two nodes in an Activity diagram. Control Flow connectors bridge the flow between Activity nodes, by directing the flow to the target node once the source node's activity is completed.

Control Flows and Object Flows can define a **Guard** and a **Weight** condition.

A **Guard** defines a condition that must be true before control passes along that activity edge. A practical example of this is where two or more activity edges (Control Flows) exit from a Decision element. Each flow should have a Guard condition that is exclusive of the other and defines which edge is taken under what conditions. The Control Flow **Properties** dialog enables you to set up Guard conditions on Control Flows

and on Object Flows.

A **Weight** defines the number of tokens that can flow along a Control or Object Flow connection when that edge is traversed. Weight can also be defined on the Control Flow and Object Flow **Properties** dialogs.

Toolbox Icon



Learn More

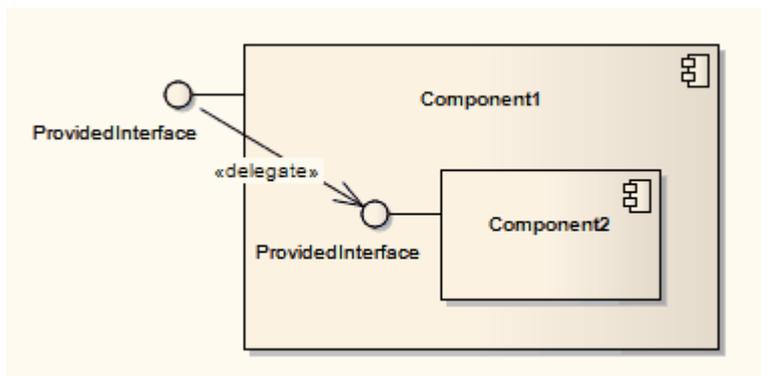
- [Activity diagram](#) ^[817]
- [Object Flow](#) ^[1003]
- [Decision Node](#) ^[886]

OMG UML specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 356*) states:

A control flow is an edge that starts an activity node after the previous one is finished.

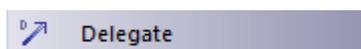
6.5.9 Delegate



Description

A Delegate connector defines the internal assembly of a component's external Ports and Interfaces, on a Component diagram. Using a Delegate connector wires the internal workings of the system to the outside world, by a delegation of the external interfaces' connections.

Toolbox Icon



Learn More:

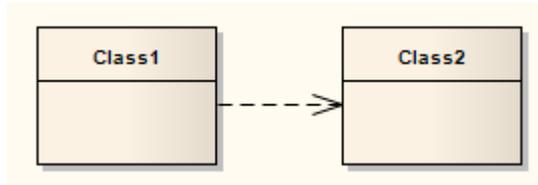
- [Component Diagram](#) ^[807]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 156*) states:

A delegation connector is a connector that links the external contract of a component (as specified by its ports) to the internal realization of that behavior by the component's parts. It represents the forwarding of signals (operation requests and events): a signal that arrives at a port that has a delegation connector to a part or to another port will be passed on to that target for handling.

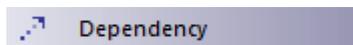
6.5.10 Dependency



Description

Dependency relationships are used to model a wide range of dependent relationships between model elements in Use Case, Activity and Structural diagrams, and even between models themselves. You can create the Dependency from the **Common** page of the **Toolbox**. The Dependencies package as defined in UML 2.3 has many derivatives, such as Realize, Deployment and Use. Once you create a Dependency you can further refine its meaning by applying a specialized stereotype.

Toolbox Icon



See Also:

- [Apply a Stereotype](#)^[979]

Learn More:

- [Realization](#)^[1007]
- [Deployment](#)^[980]
- [Use](#)^[1014]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 64*) states:

A dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation. This means that the complete semantics of the depending elements is either semantically or structurally dependent on the definition of the supplier element(s).

6.5.10.1 Apply a Stereotype

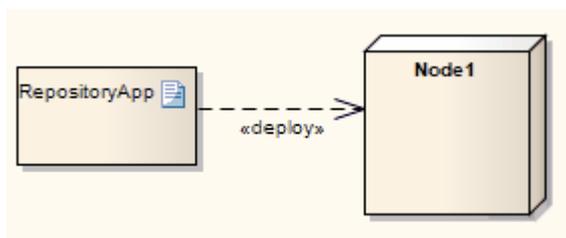
How to:

To apply a stereotype to a Dependency relationship, follow the steps below:

Step	Action
1	Select the Dependency relationship to change
2	Right-click on the connector and, from the context menu, select the Dependency Properties option The Dependency Properties dialog displays
3	In the Stereotype field, either type in the required stereotype name or click on the drop-down arrow and select the stereotype from the list
4	Click on the OK button

Alternatively, you can right-click on the Dependency relationship and select the **Advanced | Dependency Stereotypes** context menu option, then select from a shorter list of standard stereotypes.

6.5.11 Deployment



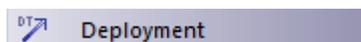
Description

A Deployment is a type of Dependency relationship that indicates the deployment of an artifact onto a node or executable target, typically in a Deployment diagram. A Deployment can be made at type and instance levels. At the type level, a Deployment would be made for every instance of the node. Deployment can also be specified for an instance of a node, so that a node's instances can have varied deployed artifacts. With composite structures modeled with nodes defined as Parts, Parts can also serve as targets of a Deployment relationship.

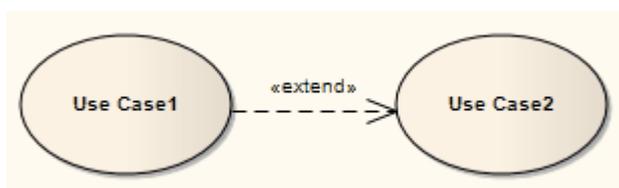
Learn More:

- [Deployment Diagram](#) ^[804]

Toolbox Icon



6.5.12 Extend



Description

An *Extend* connection is used to indicate that an element extends the behavior of another. Extensions are used in Use Case models to indicate that one Use Case (optionally) extends the behavior of another. An extending Use Case often expresses alternative flows.

Toolbox Icon



Learn More:

- [Use Case Diagram](#)

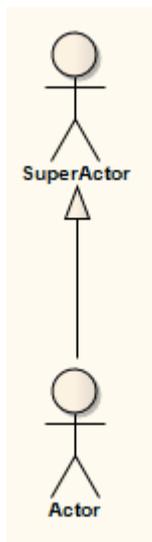
OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 587*) states:

This relationship specifies that the behavior of a Use Case may be extended by the behavior of another (usually supplementary) Use Case. The extension takes place at one or more specific extension points defined in the extended Use Case. Note, however, that the extended Use Case is defined independently of the extending Use Case and is meaningful independently of the extending Use Case. On the other hand, the extending Use Case typically defines behavior that may not necessarily be meaningful by itself. Instead, the extending Use Case defines a set of modular behavior increments that augment an execution of the extended Use Case under specific conditions.

Note that the same extending Use Case can extend more than one Use Case. Furthermore, an extending Use Case may itself be extended.

6.5.13 Generalization



Description

A Generalization is used to indicate inheritance. Drawn from the specific classifier to a general classifier, the generalization's implication is that the source inherits the target's characteristics. It is used typically in Class, Component, Object, Package, Use Case and Requirements diagrams.

You can also define template binding parameters for a Generalize connector between a binding Class and a parameterized Class.

Toolbox Icon

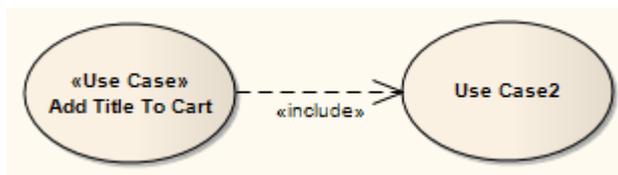


OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 73*) states:

A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. Each instance of the specific classifier is also an indirect instance of the general classifier. Thus, the specific classifier inherits the features of the more general classifier.

6.5.14 Include



Description

An Include connection indicates that the source element includes the functionality of the target element. Include connections are used in Use Case models to reflect that one Use Case includes the behavior of another. Use an Include relationship to avoid having the same subset of behavior in many Use Cases; this is similar to delegation used in Class models.

Toolbox Icon



Learn More

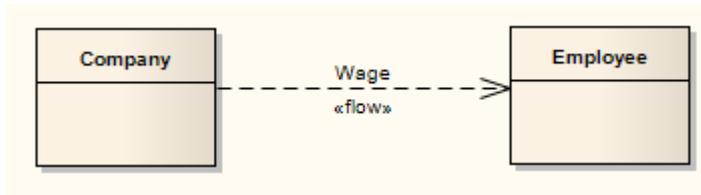
- [Use Case Diagram](#) ⁸¹³

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 591*) states:

Include is a DirectedRelationship between two Use Cases, implying that the behavior of the included Use Case is inserted into the behavior of the including Use Case. It is also a kind of NamedElement so that it can have a name in the context of its owning Use Case. The including Use Case may only depend on the result (value) of the included Use Case. This value is obtained as a result of the execution of the included Use Case.

6.5.15 Information Flow



Description

An Information Flow represents information items or classifiers flowing between two elements in any diagram. The connector is available from the Common page of the Toolbox and from every Quick Link menu.

You can have more than one Information Flow connector between the same two elements, identifying which items flow between the two under differing conditions.

Toolbox Icon



See Also

- [Using Information Flows](#) ^[983]
- [Conveying Information on a Flow](#) ^[984]
- [Realizing an Information Flow](#) ^[985]

Learn More

- [Information Item](#) ^[954]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 606*) states:

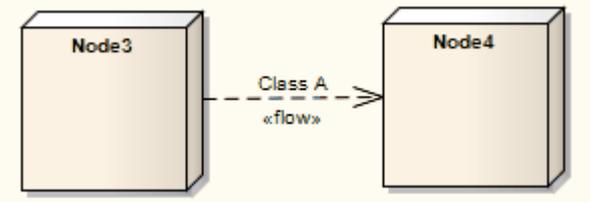
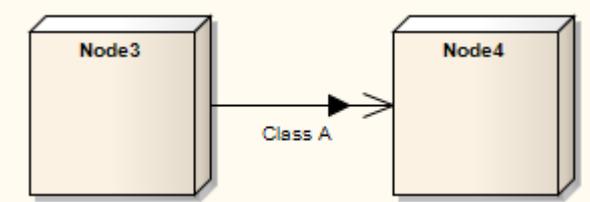
An InformationFlow specifies that one or more information items circulates from its sources to its targets.

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 607*) also states:

An information flow is an abstraction of the communication of an information item from its sources to its targets. It is used to abstract the communication of information between entities of a system. Sources or targets of an information flow designate sets of objects that can send or receive the conveyed information item.

6.5.15.1 Using Information Flows

Step	Action
1	Open a diagram and add two elements (for example, Nodes on a Deployment diagram)
2	Click on the <i>Information Flow</i> connector in the Common page of the Toolbox and drag between the two elements The Information Items Conveyed dialog displays

Step	Action
3	<p>Add the classifier or information item element(s) to the Information Flow</p> <p>The diagram now resembles the following</p> 
4	Add another connector between the same two elements (for example, a <i>Communication Path</i> connector)
5	<p>Right-click the connector and select the Advanced Information Flows Realized context menu option</p> <p>The Information Flows Realized dialog displays</p>
6	<p>Tick the checkbox against the required classifier element and click on the OK button</p> <p>The combined connector now resembles the following:</p> 

Notes:

- Once the connectors are combined, you cannot access the Information Items Conveyed dialog directly; you add or delete information items on the connector using the Information Items Realized dialog
- If you have more than one Information Flow connector between the elements, they form part of the same combined connector; you can again work on them separately through the Information Items Realized dialog
- If you have information flows in a diagram that you use as the source for a Pattern, the Information Items Conveyed and Information Flows Realized data is not copied into the Pattern
- You can locate, in the Project Browser, the classifier or information item element(s) conveyed on the Information Flow connector, using the **Find Information Item** context menu option on the connector

6.5.15.2 Convey Information on a Flow

As you create an Information Flow connector between two elements, you can specify which Information Items or classifiers are conveyed on this flow.

To specify these Information Items or classifiers, right-click on the connection and select the **Advanced | Information Item Conveyed** context menu option. The **Information Items Conveyed** dialog displays.

Button	Action
Add	Display the Select <Item> dialog, from which you select the required Information or Classifier element(s).

Button	Action
Remove	Remove the selected item.

Notes:

- If you select more than one element, they are listed in one entry for the Information Flow connector.

Learn More

- [Information Flow](#)^[983]
- [Information Item](#)^[954]

6.5.15.3 Realize an Information Flow

The **Information Flows Realized** dialog displays all flows that can be realized on the selected connector. To realize an Information Flow on this connector, select the corresponding checkbox and click on the **OK** button.

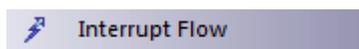
If you want to change the information items conveyed on an information flow, click on the flow *text* and click on the **Items** button. The Information Items Conveyed dialog displays, and you can add or remove items as required. When you click on the **OK** button, the **Information Items Realized** dialog redisplay and you can realize the selected flow or flows as above.

Learn More:

- [Information Flow](#)^[983]

6.5.16 Interrupt Flow**Description**

The *Interrupt Flow* is a connection used to define the two UML concepts of connectors for Exception Handler and Interruptible Activity Region. An Interrupt Flow is a type of activity edge. It is typically used in an Activity diagram.

Toolbox Icon**Learn More**

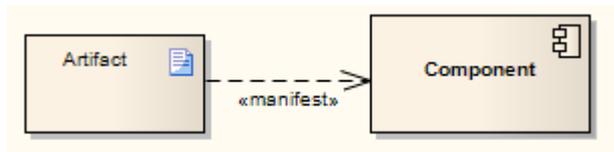
- [Exception Handler](#)^[892]
- [Interruptible Activity Region](#)^[907]
- [Activity diagram](#)^[817]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 327*) states:

An activity edge is an abstract class for directed connections between two activity nodes.

6.5.17 Manifest



Description

A Manifest relationship indicates that the Artifact source embodies the target model element, typically in Component and Deployment diagrams. Stereotypes can be added to Enterprise Architect to classify the type of manifestation of the model element.

Toolbox Icon



Learn More

- [Artifact Element](#) ^[940]
- [Component Diagram](#) ^[807]
- [Deployment Diagram](#) ^[804]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 212*) states:

An artifact embodies or manifests a number of model elements. The artifact owns the manifestations, each representing the utilization of a packageable element.

Specific profiles are expected to stereotype the manifestation relationship to indicate particular forms of manifestation, e.g. «tool generated» and «custom code» might be two manifestations for different classes embodied in an artifact.

6.5.18 Message

Messages indicate a flow of information or transition of control between elements. Messages can be used by Timing Diagrams, Sequence Diagrams and Communication Diagrams (but not Interaction Overview diagrams) to reflect system behavior. If between Classes or classifier instances, the associated list of operations is available to specify the event.

Moving a Message can disrupt the organization of other features on the diagram. To avoid this, and move *only* the Message, press **(Alt)** while you move the Message.

Toolbox Icon

 Message**See Also**

- [Message \(Sequence Diagram\)](#)^[987]
- [Message \(Communication Diagram\)](#)^[996]
- [Message \(Timing Diagram\)](#)^[999]

Learn More

- [Timing Diagram](#)^[999]
- [Sequence Diagram](#)^[987]
- [Communication Diagram](#)^[996]

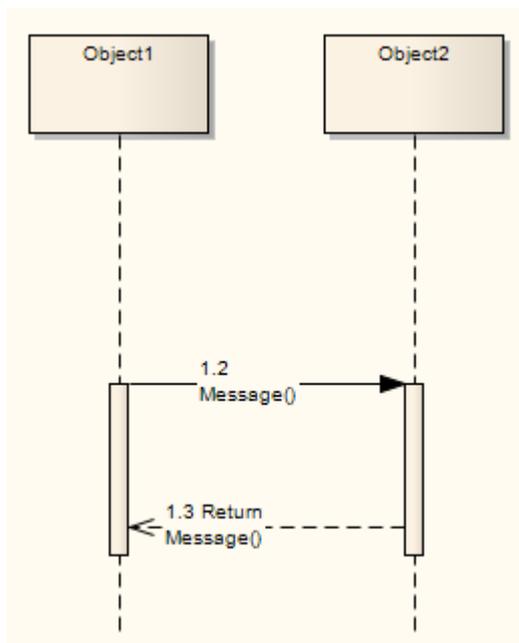
OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 491*) states:

A Message defines a particular communication between Lifelines of an Interaction.

A Message is a NamedElement that defines one specific kind of communication in an Interaction. A communication can be, for example, raising a signal, invoking an Operation, creating or destroying an Instance. The Message specifies not only the kind of communication given by the dispatching ExecutionSpecification, but also the sender and the receiver.

A Message associates normally two OccurrenceSpecifications - one sending OccurrenceSpecification and one receiving OccurrenceSpecification.

6.5.18.1 Message (Sequence Diagram)

Sequence diagrams depict work flow or activity over time using Messages passed from element to element. These Messages correspond in the software model to Class operations and behavior.

How to:

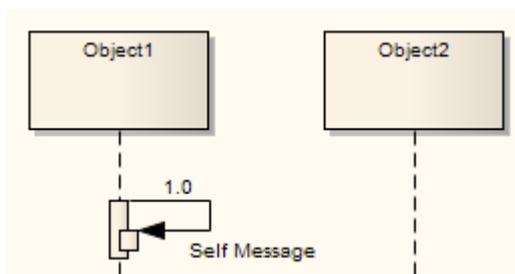
To create a Message on a Sequence diagram, follow the steps below:

Step	Action	See Also
1	<p>Access the Sequence diagram</p> <p>The Interaction pages of the Toolbox display.</p>	
2	<p>In the Interaction Relationships page, click on the Message icon, click on the source object and drag the cursor to the destination (target) object</p> <p>The Message Properties dialog displays (if not, right-click on the Message and select the Message Properties context menu option)</p>	
3	<p>In the Message field, type the Message name:</p> <ul style="list-style-type: none"> • If the Message flow is <i>towards</i> a Class element (dropped in from a Class diagram) or a Lifeline element having a classifier, and the <i>destination</i> Class has defined <i>operations</i>, you can click on the drop-down arrow and select an appropriate operation name; the Message then reflects the destination Class operations • If the available operations are not appropriate, you can click on the Operations button and define a new operation in the target element, using the Operations dialog • If you create a Message without making reference to the target Class operations, no new operation is added to the target Class 	<p>Class Element ⁹⁴</p> <p>Lifeline ⁹¹</p> <p>Operations Dialog ⁷⁰</p>
4	<p>In the Parameters field, type any parameters that the Message has, as a comma-separated list</p> <p>If required, in the Parameter Values field type the actual value for each parameter, again as a comma-separated list</p>	
5	<p>If the Message is a return message, in the Return Value field enter the returned value or type</p> <p>It is possible to depict returns from a Self Message; simply create a second Self Message at the end of execution and select the Is Return checkbox in the Control Flow Type panel</p>	<p>Self-Message ⁹⁸</p>
6	<p>If the Message flow is <i>from</i> a Class element or Lifeline element with classifier that has defined <i>attributes</i>, click on the drop-down arrow in the Assign to field and select an appropriate attribute name</p> <p>The Message reflects the attributes from the <i>source</i> Class; you cannot add further attributes to the source Class here - if no appropriate attribute is listed, open the element Properties dialog and add the required attribute</p> <p>Otherwise, if required, type the name of the object to assign the message flow to</p>	
7	<p>In the Stereotype field, type or select an optional stereotype for the connector (this is displayed on the diagram, if entered)</p>	
8	<p>If required, in the Alias field type an alias for the name of the Message</p> <p>On the diagram, the alias displays if the Use Alias if Available checkbox is selected on the Diagram tab of the Diagram Properties dialog</p> <p>The Alias displays instead of or as well as the Message name, depending on the setting selected in the Alias Usage panel of the Diagram Behavior page of</p>	<p>Diagram Behavior Options ⁴³</p>

Step	Action	See Also
	the Options dialog	
9	In the Condition field, type any conditions that must be true in order for the message to be sent	
10	In the Synch: field in the Control Flow Type panel, select Synchronous or Asynchronous as appropriate	Asynchronous Signal Message [99b]
11	In the Lifecycle field, select New to create a new element at the end of the Message, or Delete to terminate the message flow at the end of the Message If neither case applies, leave the field at the default of <none>	
12	If required, in the Notes field type any explanatory notes You can format the notes using the Notes toolbar at the top of the field	
13	Click on the OK button to save the Message definition <ul style="list-style-type: none"> You can change the timing details of a message on the Timing Details dialog, and emphasize the sequence of closely-ordered messages using General Ordering To toggle the numbering of messages on a Sequence diagram, select or deselect the Show Sequence Numbering checkbox on the Options dialog 	Change the Timing Details [99c] General Ordering [99d] Sequence diagrams [84g]

Toolbox Icon:**Learn More:**

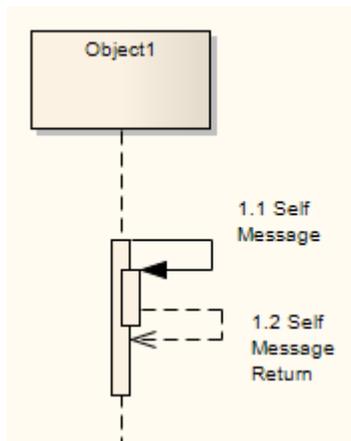
- [Call](#) [99f]
- [Message Examples](#) [99g]
- [Co-Region Notation](#) [99h]

6.5.18.1.1 Self-Message

A Self-Message reflects a new process or method invoked within the calling lifeline's operation. It is a specification of a Message, typically in a Sequence diagram.

Self-Message as Return:

It is possible to depict a return from a Self Message call.



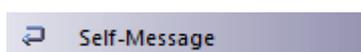
How to:

To create a Self Message return, follow the steps below:

Step	Action
1	Create a second Self Message at the end of execution
2	Double-click on the Message name to open the Message Properties dialog
3	Select the Is Return checkbox
4	Raise the Activation level of the return

Self-Message Calls indicate a nested invocation; new activation levels are added with each Call.

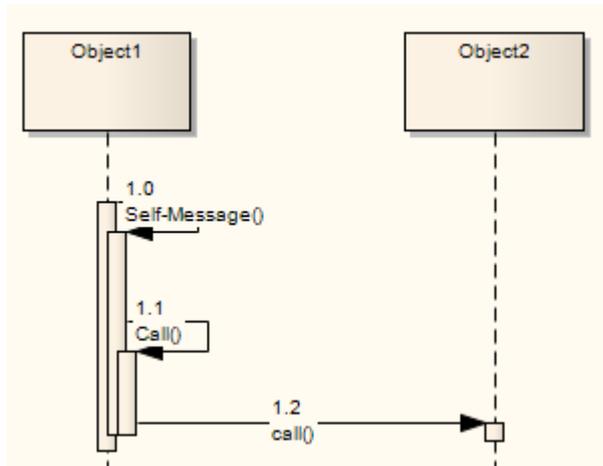
Toolbox Icon:



Learn More

- [Sequence Diagram](#) ^[849]
- [Message](#) ^[986]
- [Raise the Activation Level](#) ^[856]
- [Call](#) ^[991]

6.5.18.1.2 Call



A Call is a type of Message connector that extends the level of activation from the previous Message. All Self-Messages create a new activation level, but this focus of control usually ends with the next Message (unless activation levels are manually adjusted). Self-Message Calls, as depicted above by the first Call, indicate a nested invocation; new activation levels are added with each Call. Unlike a regular Message between elements, a Call between elements continues the existing activation in the source element, implying that the Call was initiated within the previous Message's activation scope.

Toolbox Icon

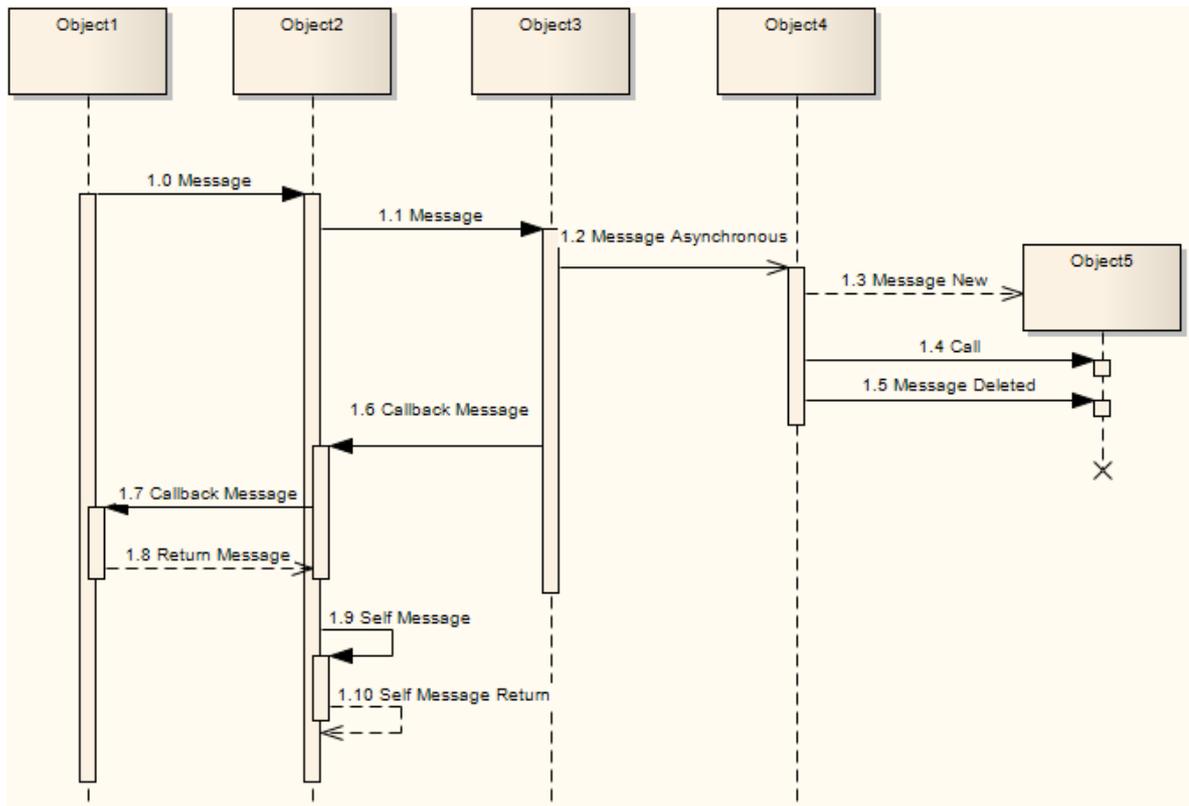


Learn More

- [Sequence Message](#)^[987]
- [Self-Message](#)^[989]
- [Sequence Element Activation](#)^[855]

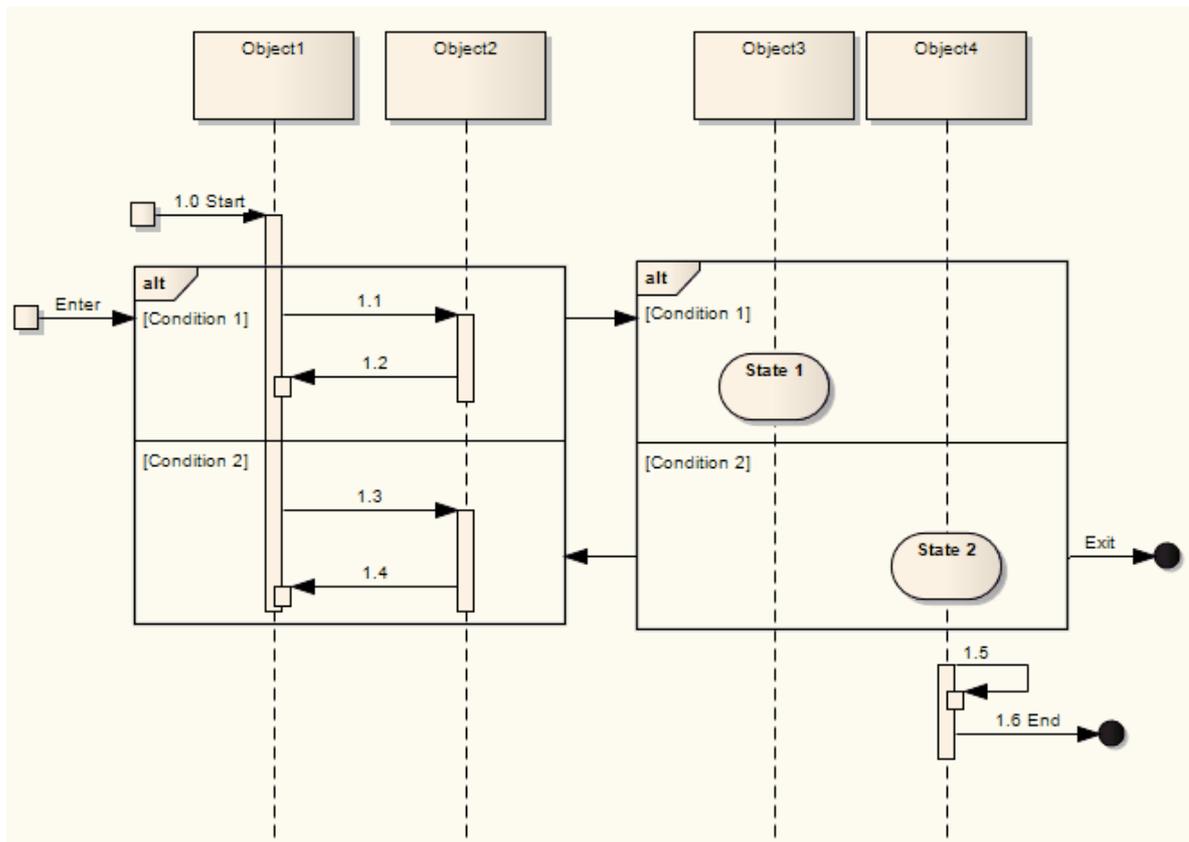
6.5.18.1.3 Message Examples

The following are different types of Messages available on Sequence Diagrams. Note that Messages on Sequence diagrams can also be modified with Shape Scripts.



Other Sequence Messages

The following are examples of Messages that are not part of the sequence described by the diagram.



Learn More

- [Messages](#) ^[987]
- [Sequence Diagrams](#) ^[849]
- [Shape Scripts](#) ^[1089]

6.5.18.1.4 Change the Timing Details

It is possible to change the timing details of a Message in a Sequence diagram by right-clicking on the Message connector and selecting the **Timing Details** context menu option. The **Timing Details** dialog displays.

Option	Action
Duration Constraint	Indicate the minimum and maximum limits on how long a message can last.
Duration Constraint Between Messages	Indicate the minimum and maximum interval between sending or receipt of the previous message at the current message's source Lifeline, and sending the current message.
Duration Observation	Capture the duration of a message.
Timing Constraint	Indicate the minimum and maximum time at which the message should arrive at the target.
Timing Observation	Capture the point at which the message was sent.

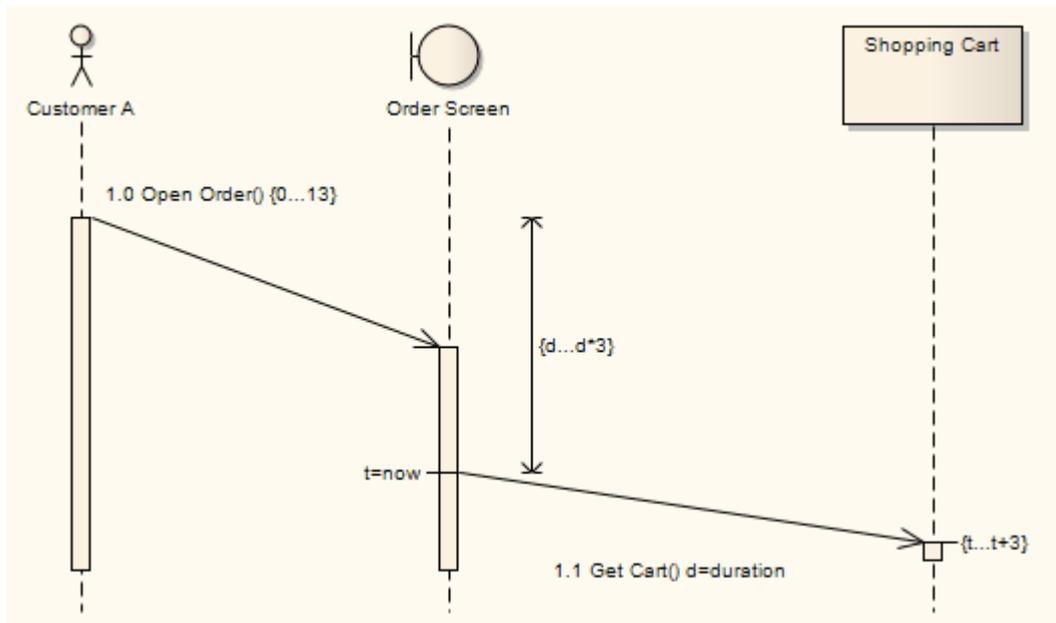
See the OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 511*).

In the diagram below, on the *Open Order Message*:

- **Duration Constraint** has been set to **0...13**.

On the *Get Cart* Message:

- **Duration Constraint Between Messages** has been set to **d...d*3**
- **Duration Observation** has been set to **d=duration**
- **Timing Constraint** has been set to **t...t+3**
- **Timing Observation** has been set to **t=now**



By typing a value in the **Duration Constraint** field, you enable the Message angle to be adjusted. After clicking on the **OK** button on the **Timing Details** dialog, click on the head of the Message connector and drag the connector up or down to change the angle. You cannot extend the angle beyond the life line of the connecting sequence object or create an angle of less than 5 degrees.

You can also create the **Duration Constraint Between Messages** line by dragging the General Ordering arrow up to the point at which the previous message joins the source Lifeline for the current message. A dialog displays on which you enter the value for the constraint. Having created the line, you can move it to any point within half way along the current message and half way along the previous message, to avoid overlap with other message timing details. You can edit or delete the value either through the **Timing Details** dialog or by right-clicking on the line itself and selecting the appropriate context menu option.

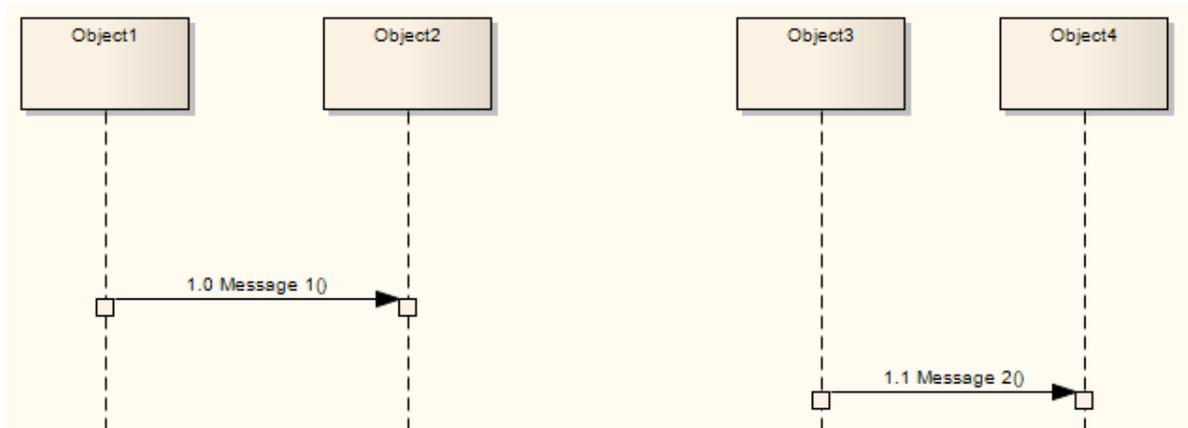
Learn More

- [Sequence Diagram](#) ⁸⁴⁹
- [Message](#) ⁹⁸⁶
- [General Ordering](#) ⁹⁹⁴

6.5.18.1.5 General Ordering

In a Sequence diagram, the workflow is represented by the sequence of Messages down the diagram. Messages near the top of the diagram are passed before Messages lower down the diagram.

Consider the following diagram.

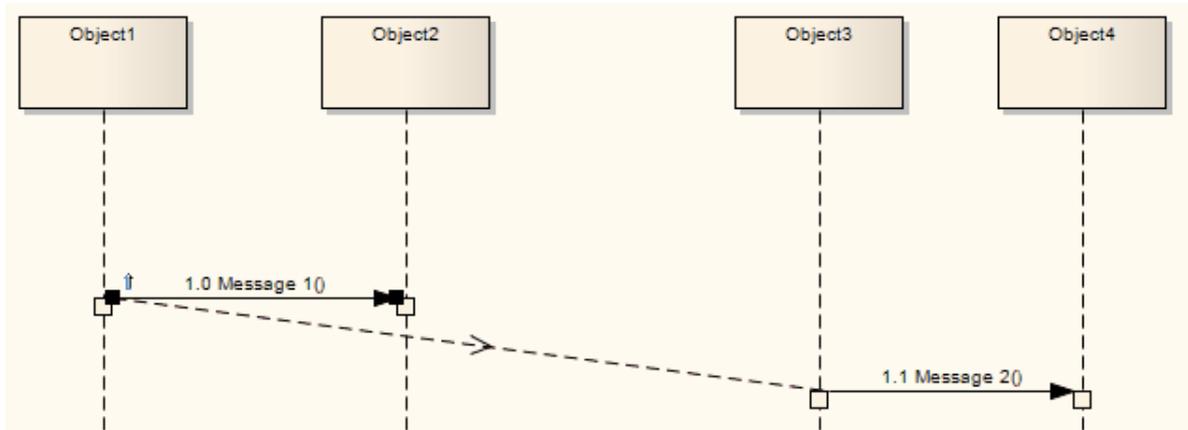


Message 1 is earlier than Message 2. However, in a complex diagram, or when representing finely timed operations or parallel processing, this might not be apparent. You can reinforce the sequence using a General Ordering arrow.

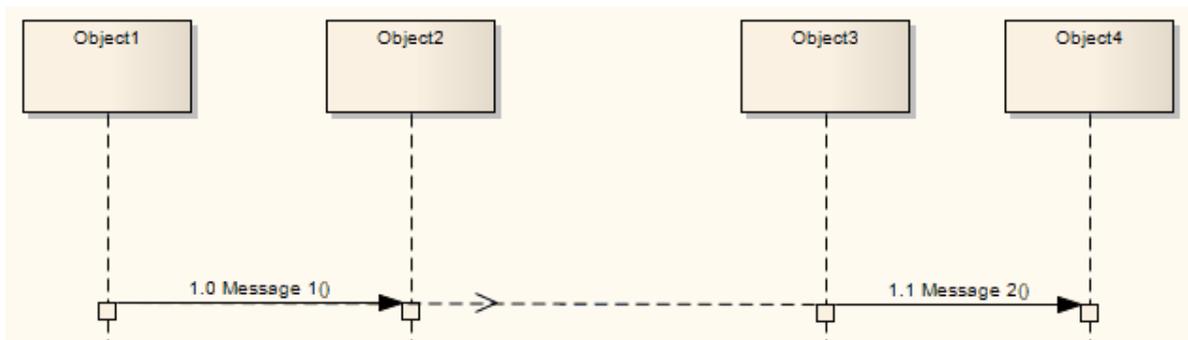
Click on the Message arrow. A small arrow displays at the source anchor point.



Click on this arrow and drag it to the start of the next Message in sequence (Message 2 in the example). The General Ordering arrow displays, indicating that the second Message follows the first.



The General Ordering arrow is exaggerated in the above figure. You would normally have the arrow running almost horizontal across the diagram.



You can have more than one General Ordering arrow issuing from or targeting a Message, if necessary.

6.5.18.1.6 Asynchronous Signal Message

You define a Message as an asynchronous signal message by displaying the **Message Properties** dialog and setting the **Synch** field to **Asynchronous** and the **Kind** field to **Signal**. A *synchronous* message cannot be used to convey signals, so setting the **Synch** field to **Synchronous** disables the **Kind** field.

Return Value, **Assign To** and the **Operations** button, which are not applicable to asynchronous *signals*, are disabled.

The **Operations** button changes to a **Signal** button, which you click on to associate the asynchronous signal message with a Signal element in the model. You can type the arguments corresponding to the Signal attributes into the **Argument(s)** field.

When you click on the **Signal** button, the **Select *Signal** dialog displays, through which you locate and select the required Signal element. (The **Select *Signal** dialog is a variation of the **Select <Item>** dialog.)

Learn More

- [Message Properties Dialog](#)^[987]

6.5.18.1.7 Co-Region Notation

Co-Region notation can be used as a short hand for parallel combined fragments. To access the **Co-Region** submenu, right-click on a connector in a Sequence diagram and select the **Co-Region** context menu option. There are four sub-options available:

- **Start at head**
- **End at head**
- **Start at tail**
- **End at tail.**

6.5.18.2 Message (Communication Diagram)

A Message in a Communication diagram is equivalent in meaning to a Message in a Sequence diagram. It implies that one object uses the services of another object, or sends a message to that object. Communication Messages in Enterprise Architect are always associated with an Association connector between object instances. Always create the Association first, then add a Message to the connector.

Messages can be dragged into a suitable position by clicking and dragging on the message text.

Communication Messages are ordered to reflect the sequencing of the diagram. The numbering scheme should reflect the nesting of each event. A sequencing scheme could be:

```
1
2, 2.1, 2.2, 2.3
3.
```

This would indicate the single sequence of events 2.1, 2.2 and 2.3 occurs within an operation initiated by event 2. This is the default pattern applied by Enterprise architect

Alternatively, the sequence could be:

```
1
2, 2.1, 2.1.1, 2.1.1.1
  2.2, 2.2.1, 2.2.1.1
3
```

This would indicate that two sequences of events can be initiated by event 2, and 2.1 and 2.2 are separate

sequences, not consecutive events in one sequence. You can set the sequence pattern and order using the **Message Properties** dialog and the **Sequence Communications** dialog.

If the target object is a Class or has its instance classifier set, the drop-down list of possible message names includes the exposed operations for the base type.

Learn More:

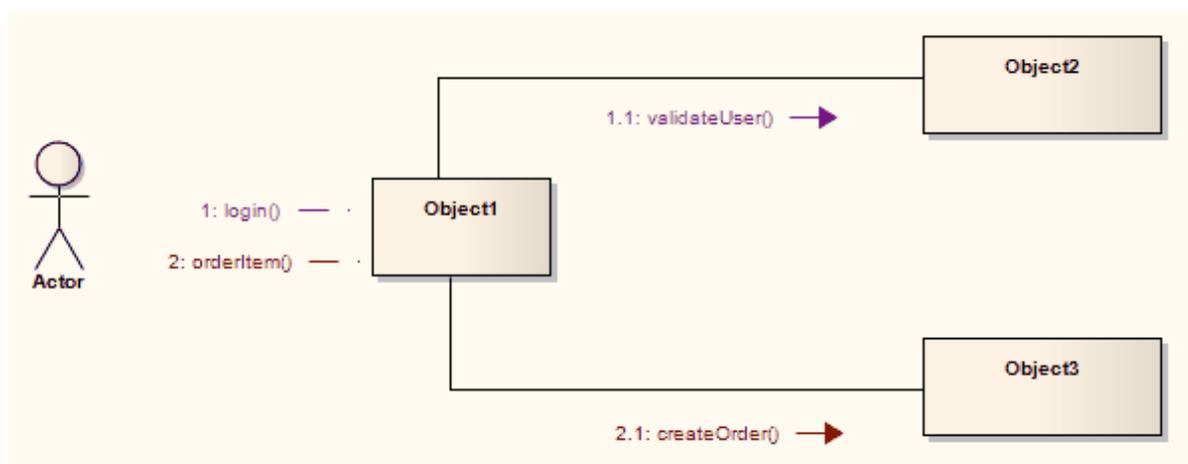
- [Communication Diagram](#) ^[859]
- [Create a Communication Message](#) ^[997]
- [Re-Order Messages](#) ^[998]

6.5.18.2.1 Create a Communication Message

How to:

To create a Communication Message, follow the steps below

Step	Action
1	Open a diagram (one of: Communication, Analysis, Interaction Overview, Object, Activity or State Machine)
2	Add the required objects
3	Add an Association relationship between each pair of objects that communicate
4	Right-click on an Associate to display the context menu
5	Select the option to add a Message from one object to the other
6	When the Message Properties dialog displays, type in a name and any other required details
7	Click on the OK button The Message is added, connected to the Association and Object instances
8	Move the Message to the required position



Learn More:

- [Communication Message](#)^[996]
- [Message Properties Dialog](#)^[987]

6.5.18.2.2 Re-Order Messages

When constructing your Communication diagram, it is frequently necessary to create or delete Message 'groups' and to re-order the sequence of Messages. There are two dialogs that help you perform these tasks: the Message Properties dialog and the Sequence Communications dialog.

Organize Message Groups:

If you have several Messages in the form 1.1, 1.2, 1.3, 1.4, for example, but would like to start a new numbering group on, say, the third Message (that is, 1.1, 1.2, **2.1**, 2.2, 2.3), you can change a Message in the series to a *Start Group* message.

How to:

To reorganize message groups, follow the steps below:

Step	Action
1	Double-click on a Message <i>name</i> The Message Properties dialog displays
2	To make the selected Message the start of a new group, select the Start New Group checkbox
3	If required, in the Notes field, type an explanatory note You can format the text using the Notes toolbar at the top of the field
4	Click on the OK button to save changes

Sequence Messages:

In larger and more complex diagrams, you might have to use deeper levels of Messages in a group; for example, 1, 1.2, 1.2.1, 1.2.1.1. You might also have to change the sequence of Messages, making Message 1.3, for example, into Message 1.1.

How to:

To change the sequence or level of Messages, follow the steps below:

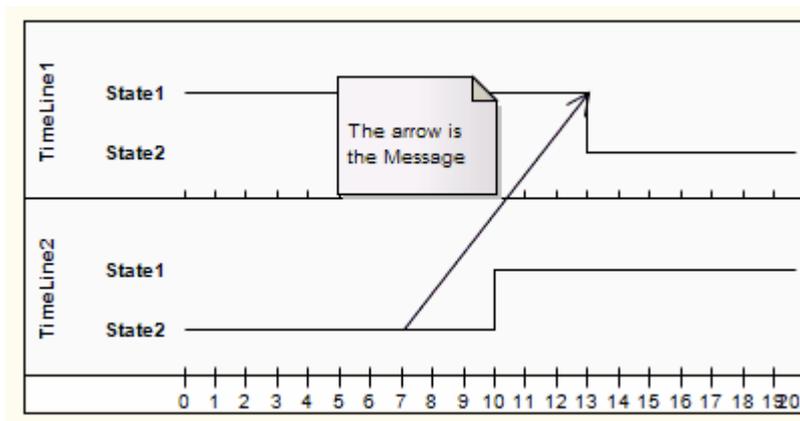
Step	Action
1	Either: <ul style="list-style-type: none"> • Select the Diagram Advanced Sequence Messages menu option • Click on the diagram background and select the Sequence Communication Messages context menu option or • Right-click on a Message and select the Sequence Communication Messages

Step	Action
	context menu option The Communication Messages dialog displays
2	Click on the Message to adjust and, at the bottom of the dialog, click on the: <ul style="list-style-type: none"> 'Up Hand' or 'Down Hand' buttons to move the Message up or down the sequence (e.g. Message 1.2 to Message 1.1 or 1.3) 'Left Hand' or 'Right Hand' buttons to move the Message up or down a level (e.g. Message 1.2.1 to Message 1.2 or Message 1.2.1.1)
3	Repeat step 2 until the Message sequence and levels match your requirements You might have to adjust other Message numbers (in group, sequence or level) to accommodate the changes you have made
4	Click on the OK button to save changes

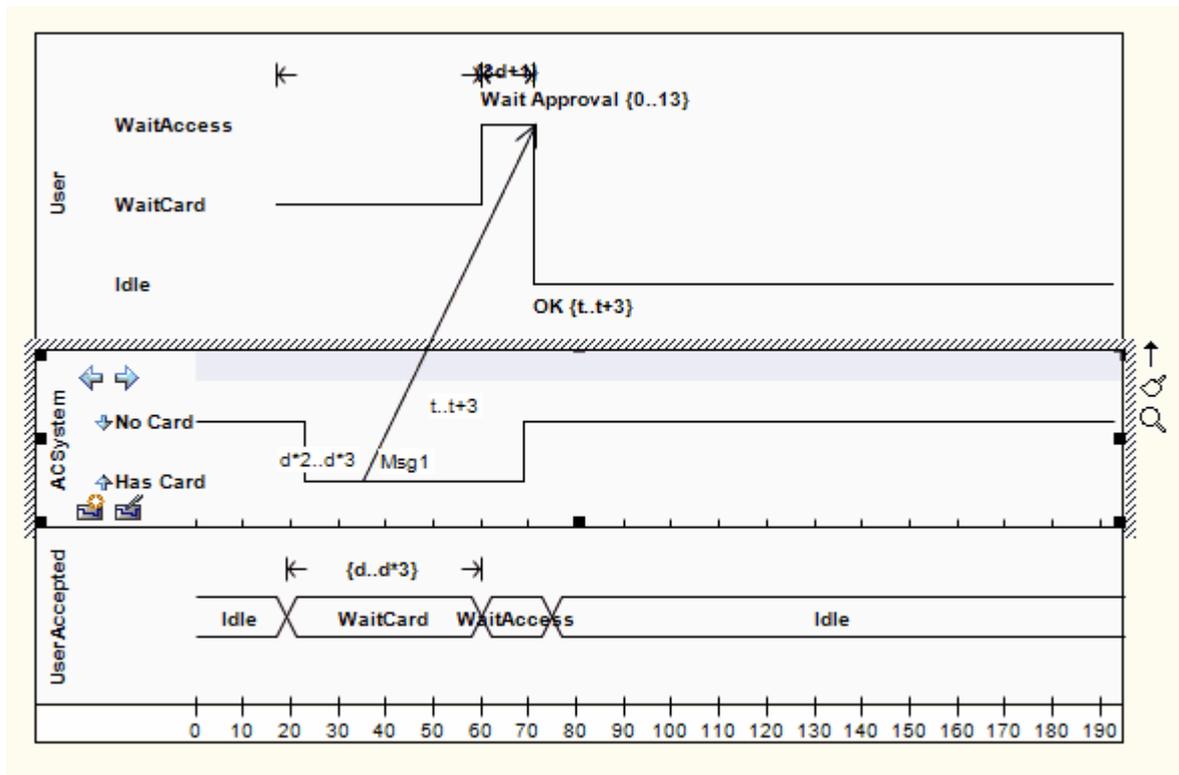
Learn More

- [Communication Message](#) ⁹⁹⁶

6.5.18.3 Message (Timing Diagram)



Messages are the communication links between Lifelines in a Timing diagram. In the case of a Timeline, a Message is a connection between two Timeline objects.



See *UML Superstructure Specification, v2.1.1, figures 14.30 and 14.31, p. 520.*

Toolbox Icon



Learn More

- [Timing Diagram](#)^[830]
- [Lifeline](#)^[910]
- [Create a Timing Message](#)^[1000]

6.5.18.3.1 Create a Timing Message

To create a Message in a Timing diagram, at least two Lifeline objects (State or Value) must be created first, each with existing transition points.

How to:

To create a Message between Lifelines, follow the steps below:

Step	Action
1	Click on one of the Lifelines in the Timing diagram
2	Select the Message icon from the Timing Relationships page of the Toolbox (More tools Timing)
3	Drag the cursor onto the Lifeline at the point at which the Message originates

Step	Action
	The Timing Message dialog displays (if not, double-click on the Message)
4	The dialog consists of a set of transition points Each transition point can be defined with the properties listed in the following table

Property	Description
Start	Defines the lifeline where the message originates
End	Defines the lifeline where the message terminates

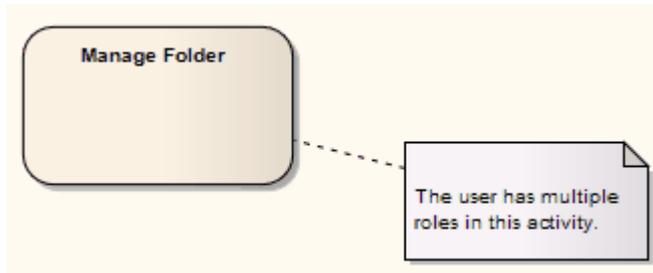
The following properties are set by default when a Message is created by dragging the cursor between two Lifelines.

Property	Description
Start Time	Specifies the start time for a message
End Time	Specifies the end time for a message
Name	The name of the message
Time Observation	Provides information on the time of a sent message
Duration Observation	Indicates the interval of a Lifeline at a particular state, begun from a message receipt
Transition To	The state in the target Lifeline that the Message points to
Event	The occurring event
Time Constraint	The time taken to transmit a message
Duration Constraint	Pertains to a lifeline's period at a particular state The constraint could be instigated by that Lifeline's receipt of a message

The following diagram shows an example of a configured Message:

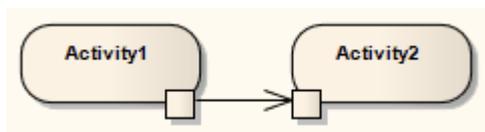
Toolbox Icon:**Learn More:**

- [Package Diagram](#) ¹⁷⁹⁶

6.5.20 Notelink**Description**

A Notelink connector connects a Note to one or more other elements of any other type.

Both Note and Notelink are available in any category of the **Toolbox**, in the **Common** page. You can also select them from the **UML Elements** toolbar.

Toolbox Icon**6.5.21 Object Flow****Description**

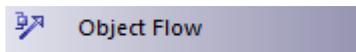
Object Flows are used in Activity diagrams and State Machine diagrams. When used in an Activity diagram, an Object Flow connects two elements, with specific data passing through it. To view sample Activity diagrams using Object Flows, see the Object Flows in Activity Diagrams topic.

In State Machine diagrams, an Object Flow is a specification of a state flow or transition. It implies the passing of an Object instance between elements at run-time.

You can insert an Object Flow from the **State** or **Activity** pages of the **Toolbox**, or from the drop-down list of all relationships located in the header toolbar. You can also modify a transition connection to an Object Flow by selecting the **ObjectFlow** checkbox on the connection **Properties** dialog.

See the Control Flow topic for information on setting up Guards and Weights on Object Flows.

Toolbox Icon



See Also

- [Object Flows in Activity Diagrams](#)^[1004]

Learn More

- [Activity Diagram](#)^[817]
- [Control Flow](#)^[977]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 389*) states:

An object flow is an activity edge that only passes object and data tokens.

6.5.21.1 Object Flows in Activity Diagrams

In Activity diagrams, there are several ways to define the flow of data between objects.

The following diagram depicts a simple Object Flow between two actions, Fill Order and Ship Order, both accessing order information.



See *UML Superstructure Specification, v2.1.1, figure 12.110, p. 391*.

This explicit portrayal of the data object *Order*, connected to the Activities by two Object Flows, can be refined by using the following format. Here, Action Pins are used to reflect the order.



See *UML Superstructure Specification, v2.1.1, figure 12.110, p. 391*.

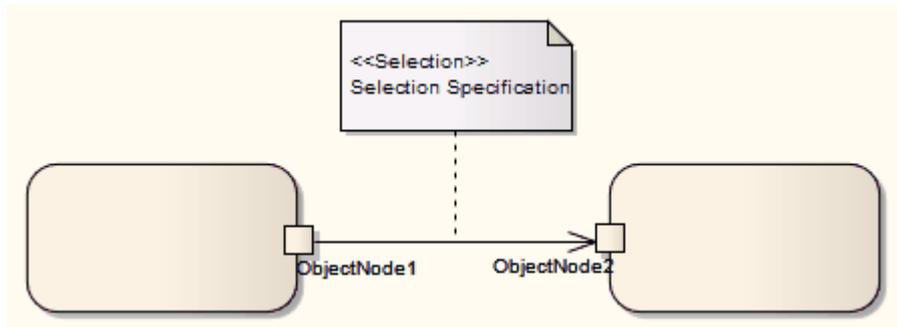
The following diagram is an example of multiple Object Flows exchanging data between two actions.



See *UML Superstructure Specification, v2.1.1, figure 12.111, p. 391.*

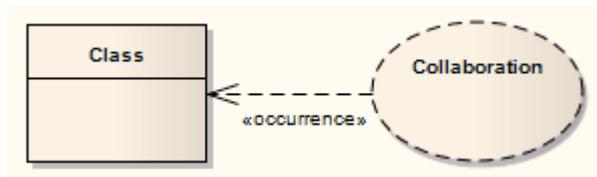
Selection and transformation behavior, together composing a sort of query, can specify the nature of the Object Flow's data access. Selection behavior determines which objects are affected by the connection. Transformation behavior might then further specify the value of an attribute pertaining to a selected object.

Selection and transformation behaviors can be defined by attaching a note to the Object Flow. To do this, right-click on the Object Flow and select the **Attach Note or Constraint** context menu option. A dialog lists other flows in the diagram, to which you can select to attach the note, if the behavior applies to multiple flows. To comply with UML 2, preface the behavior with the notation «selection» or «transformation».



See *UML Superstructure Specification, v2.1.1, figure 12.112, p. 392.*

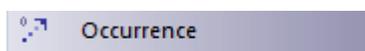
6.5.22 Occurrence



Description

An *Occurrence* relationship indicates that a Collaboration represents a classifier, in a Composite Structure diagram. An Occurrence connector is drawn from the Collaboration to the classifier.

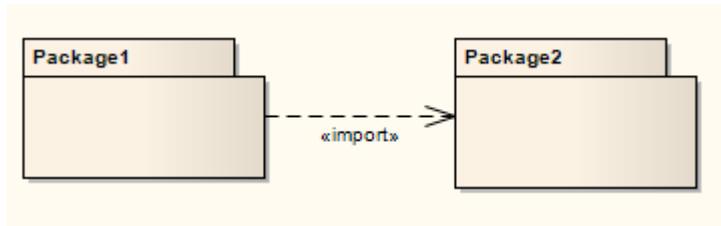
Toolbox Icon



Learn More

- [Collaboration](#)^[945]
- [Composite Structure Diagram](#)^[807]

6.5.23 Package Import



Description

A Package Import relationship is drawn from a source Package to a Package whose contents are to be imported. Private members of a target Package cannot be imported. The relationship is typically used in a Package diagram.

Toolbox Icon



Learn More

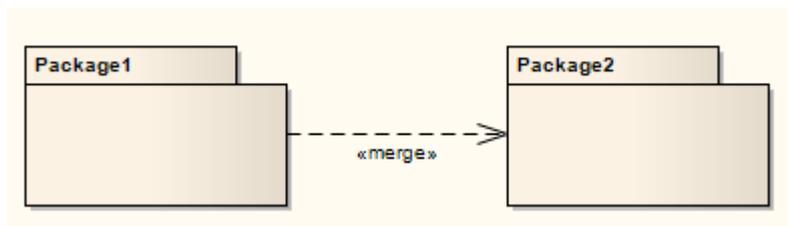
- [Package](#)^[960]
- [Package Diagram](#)^[796]

OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 112*) states:

A package import is a relationship between an importing namespace and a package, indicating that the importing namespace adds the names of the members of the package to its own namespace. Conceptually, a package import is equivalent to having an element import to each individual member of the imported namespace, unless there is already a separately-defined element import.

6.5.24 Package Merge



Description

In a Package diagram, a Package Merge indicates a relationship between two Packages whereby the contents of the target Package are merged with those of the source Package. Private contents of a target Package are not merged. The applicability of a Package Merge addresses any situation where multiple packages contain identically-named elements, representing the same thing. A Package Merge merges all matching elements across its merged Packages, along with their relationships and behaviors. Note that a Package Merge essentially performs generalizations and redefinitions of all matching elements, but the merged Packages and their independent element representations still exist and are not affected.

The Package Merge serves a graphical purpose in Enterprise Architect, but creates an ordered Package relationship applied to related Packages (which can be seen under the **Link** tab in the Package's **Properties** dialog). Such relationships can be reflected in XML exports or Enterprise Architect Automation Interface scripts for code generation or other Model Driven Architecture (MDA) interests.

Package Merge relationships are useful to reflect situations where existing architectures contain functionalities involving like elements, which are merged in a developing architecture. Merging doesn't affect the merged objects, and supports the common situation of product progression.

Toolbox Icon



Learn More

- [Package Diagram](#) ^[796]
- [Package](#) ^[960]

OMG UML Specification

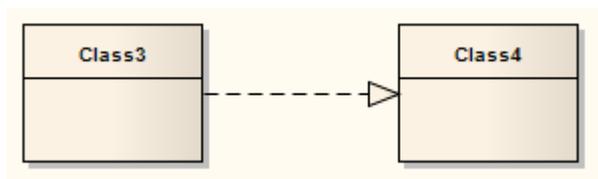
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 113-114*) states:

A package merge is a directed relationship between two packages that indicates that the contents of the two packages are to be combined. It is very similar to Generalization in the sense that the source element conceptually adds the characteristics of the target element to its own characteristics resulting in an element that combines the characteristics of both.

This mechanism should be used when elements defined in different packages have the same name and are intended to represent the same concept. Most often it is used to provide different definitions of a given concept for different purposes, starting from a common base definition. A given base concept is extended in increments, with each increment defined in a separate merged package. By selecting which increments to merge, it is possible to obtain a custom definition of a concept for a specific end. Package merge is particularly useful in meta-modeling and is extensively used in the definition of the UML metamodel.

Conceptually, a package merge can be viewed as an operation that takes the contents of two packages and produces a new package that combines the contents of the packages involved in the merge. In terms of model semantics, there is no difference between a model with explicit package merges, and a model in which all the merges have been performed.

6.5.25 Realization



Description

A source object implements or Realizes its destination object. Realize connectors are used in a Use Case, Component or Requirements diagram to express traceability and completeness in the model. A business process or Requirement is realized by one or more Use Cases, which in turn are realized by some Classes, which in turn are realized by a Component, and so on. Mapping Requirements, Classes and such across the design of your system, up through the levels of modeling abstraction, ensures the big picture of your system remembers and reflects all the little pictures and details that constrain and define it.

You can also define template binding parameters for a Realize connector between a binding Class and a parameterized Class.

Toolbox Icon

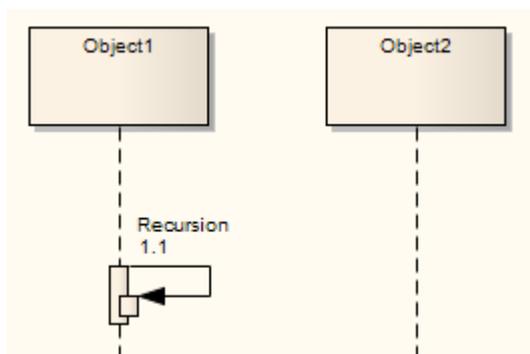


OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 131*) states:

A Realization signifies that the client set of elements are an implementation of the supplier set, which serves as the specification. The meaning of 'implementation' is not strictly defined, but rather implies a more refined or elaborate form in respect to a certain modeling context. It is possible to specify a mapping between the specification and implementation elements, although it is not necessarily computable.

6.5.26 Recursion



Description

A Recursion is a type of Message used in Sequence diagrams to indicate a recursive function.

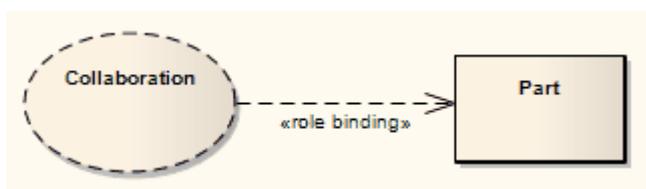
Toolbox Icon



Learn More

- [Message](#) ^[987]
- [Sequence Diagram](#) ^[849]

6.5.27 Role Binding



Description:

Role Binding is the mapping between a Collaboration Use's internal roles and the respective Parts required to implement a specific situation, typically in a Composite Structure diagram. The associated Parts can have properties defined to enable the binding to occur, and the Collaboration to take place.

A Role Binding connector is drawn between a Collaboration and the classifier's fulfilling roles, with the Collaboration's internal binding roles labeled on the classifier end of the connector.

Toolbox Icon:



Learn More:

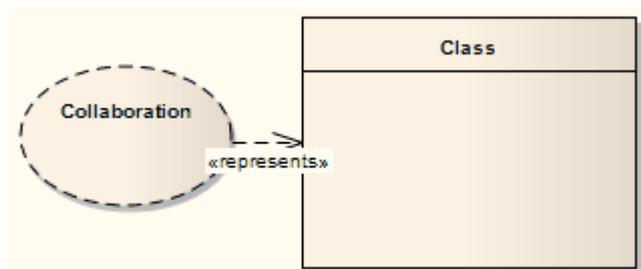
- [Collaboration Use](#) [94b]
- [Composite Structure Diagram](#) [80f]
- [Collaboration](#) [94b]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 174*) states:

A mapping between features of the collaboration type and features of the classifier or operation. This mapping indicates which connectable element of the classifier or operation plays which role(s) in the collaboration. A connectable element may be bound to multiple roles in the same collaboration use (that is, it may play multiple roles).

6.5.28 Represents



Description:

The Represents connector indicates that a Collaboration is used in a classifier, typically in a Composite Structure diagram. The connector is drawn from the Collaboration to its owning classifier.

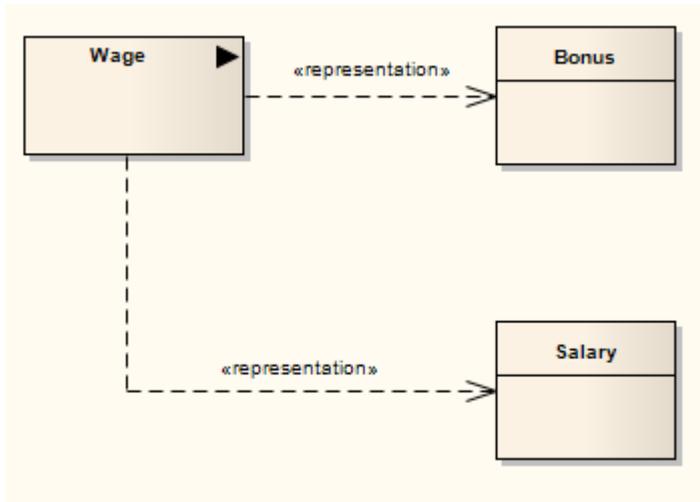
Toolbox Icon:



Learn More:

- [Collaboration](#) [94b]
- [Composite Structure Diagram](#) [80f]

6.5.29 Representation



Description:

The Representation relationship is a specialization of a Dependency, connecting Information Item elements that represent the same idea across models, typically in an Analysis diagram. For example, Bonus and Salary are both a representation of the Information Item Wage.

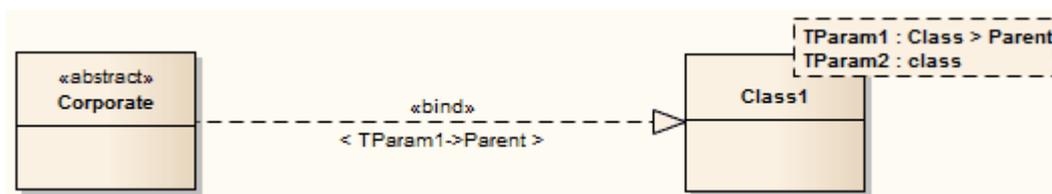
Toolbox Icon:



Learn More:

- [Dependency](#) ^[979]
- [Information Item](#) ^[954]
- [Analysis diagram](#) ^[1187]

6.5.30 Template Binding



Description:

You create a Template Binding connector between a binding Class and a parameterized Class. You then define a binding expression on that connector. However, if the binding Class requires a Generalization, Realization or Association relationship with the parameterized Class, you can define the binding expression on that relationship instead.

You can create a Template Binding connector using:

- The **Template Binding** icon on the Class Relationships page of the Diagram Toolbox
- The Quicklinker arrow next to the source Class element
- The Templates tab of the binding Class element Properties dialog; here, you create the Template Binding relationship by clicking the **Add** button under the Binding(s) panel, specifying the connector type, and selecting the target parameterized Class from the Select <Item> dialog

Each of these methods creates the connector itself. For the first two methods you then double-click on the connector to display the connector Properties dialog, on which you select the Binding tab to define *parameter substitutions* as the binding expression. The third method takes you to this dialog and tab automatically.

Toolbox Icon:



Learn More:

- [Parameterized Class](#) ^[943]
- [Parameter Substitution](#) ^[1011]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 622*) states:

A template is a parameterized element ... used to generate other model elements using TemplateBinding relationships. The template parameters for the template signature specify the formal parameters that will be substituted by actual parameters (or the default) in a binding.

6.5.30.1 Parameter Substitution

Once a Template Binding (or other binding) relationship exists, you can add parameter substitutions to identify the formal parameters that are replaced, and the actual parameters that replace them, in the binding expression.

How to:

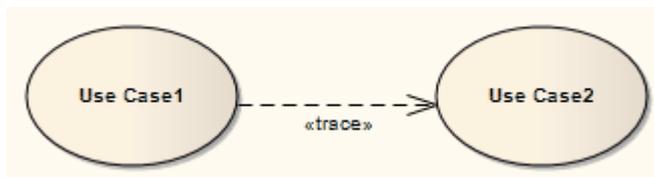
To define a parameter substitution, follow the steps below:

Step	Action
1	Display the Binding tab, in the Template Binding connector Properties dialog The Target field identifies the target parameterized Class
2	Click on the Add button below the Parameter Substitution(s) panel The next available row in the panel is enabled for editing, and the word <i><none></i> is displayed in the Formal column
3	Click on the field and on the drop-down arrow that is now displayed A list of the template parameters from the target Class displays; click on the required parameter

Step	Action
4	<p>Click on the (...) button in the corresponding Actual field for the parameter</p> <p>If the template parameter:</p> <ul style="list-style-type: none"> Does not have a constraint, a short context menu displays offering the choice of typing a free-text value into the Actual field, or selecting a classifier from the Select Classifier dialog Has a constraint defined, the Select Classifier dialog displays automatically, showing the available classifiers
5	<p>Locate and select the required classifier to replace the parameter in the binding expression</p> <p>If you do not define an Actual classifier <i>and</i> the template parameter has a default value defined, that default is used in the expression</p>
6	To edit existing parameter substitutions, click on them and make the required changes as indicated in steps 3 and 4
7	<p>Click on the Apply and/or OK button</p> <p>The parameter substitutions display as a label underneath the connector</p>

Learn More:

- [Template Binding](#)^[1010]

6.5.31 Trace**Description:**

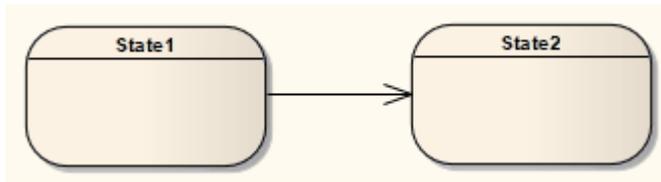
The Trace relationship is a specialization of a Dependency, connecting model elements or sets of elements that represent the same concept across models. Traces are often used to track requirements and model changes, typically in a Traceability diagram, or in a Class, Use Case, Object or Composite Structure diagram.

As changes can occur in both directions, the order of this Dependency is usually ignored. The relationship's properties can specify the trace mapping, but the trace is usually bi-directional, informal and rarely computable.

Toolbox Icon:**Learn More:**

- [Dependency](#)^[979]
- [Traceability Diagram](#)^[505]

6.5.32 Transition



Description:

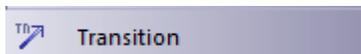
A *Transition* defines the logical movement from one State to another, in a State Machine diagram. The Transition can be controlled through the following connector Properties dialog fields:

Field	Action	See also
Guard	Type in the expression that is to be evaluated after an Event is dispatched, but before the corresponding Transition is triggered If the guard is true at that time, the Transition is enabled; otherwise, it is disabled	
Effect is a Behavior	Convert the Effect field from a free-text field to the definition of a specific Activity or behavior Enterprise Architect displays the Select <Item> dialog to prompt you to select the Activity or behavior element from the model	Select <Item> Dialog ^[690]
Effect	Either: <ul style="list-style-type: none"> Type a description of the effect of the Transition, or If you have selected the Effect is a Behavior check box, select an Activity or behavior to be performed during the Transition (to change this subsequently, click on the (...) button to redisplay the Select <Item> dialog) 	
Trigger		
Name	Specify the name of the trigger; either: <ul style="list-style-type: none"> Type the name, or Select an existing trigger in the model using the Select <Item> dialog, which you display by clicking on the (...) button 	
Type	Specify the type of trigger: Call , Change , Signal or Time : <ul style="list-style-type: none"> Call - specifies that the event is a CallEvent, which sends a message to the associated object by invoking an operation Change - specifies that the event is a ChangeEvent, which indicates that the transition is the result of a change in value of an attribute Signal - specifies that the event is a SignalEvent, which corresponds to the receipt of an asynchronous signal instance Time - corresponds to a TimeEvent; which specifies a moment in time Code generation for State Machines currently supports Change and Time trigger events only, and expects a specification value	
Specification	Specify the event instigating the Transition; either:	

Field	Action	See also
	<ul style="list-style-type: none"> Type the event (time or change), or Select an existing specification in the model using the Select <Item> dialog, which you display by clicking on the (...) button 	
New	Clear the fields ready to begin defining a new trigger	
Save	Save the newly created or edited trigger	
Delete	Remove the selected trigger from the list	
<trigger list>	List the existing triggers, which might or might not have names and types, and which can include Call and Signal triggers created in older models	

Notes:

- Fork and Join segments can have neither triggers nor guards
- You can identify hidden triggers and locate triggers in the Project Browser, using the **Find Triggers Associated** option on the Transition connector context menu - if one trigger exists for the Transition, it is immediately highlighted in the Project Browser; if more than one trigger exists, the Element Usage dialog displays; select the required trigger and click on the **Open** button to highlight the trigger in the Project Browser

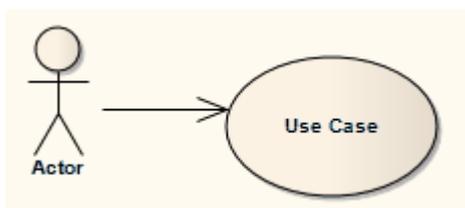
Toolbox Icon:**Learn More:**

- [State Machine Diagrams](#) ⁸¹⁵

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 568*) states:

A transition is a directed relationship between a source vertex and a target vertex. It may be part of a compound transition, which takes the state machine from one state configuration to another, representing the complete response of the state machine to an occurrence of an event of a particular type.

6.5.33 Use**Description:**

A Use relationship indicates that one element requires another to perform some interaction. The Use (or

Usage) relationship does not specify how the target supplier is used, other than that the source client uses it in definition or implementation. A Use relationship is a sub-typed Dependency relationship.

You typically use the Use relationship in Use Case diagrams to model how Actors use system functionality (Use Cases), or to illustrate usage dependencies between Classes or Components.

Notes:

- It is more usual (and correct UML) to have an Association between an Actor and a Use Case
- To depict a usage dependency on a Class or Component diagram, draw a Dependency connector; right-click on the Dependency, and select the **Dependency Stereotypes | Use** context menu option

Toolbox Icon:



Learn More:

- [Use Case Diagram](#) ^[813]
- [Association](#) ^[970]

OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 138*) states:

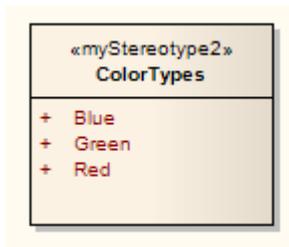
A usage is a relationship in which one element requires another element (or set of elements) for its full implementation or operation. In the metamodel, a Usage is a Dependency in which the client requires the presence of the supplier.

6.6 UML Stereotypes

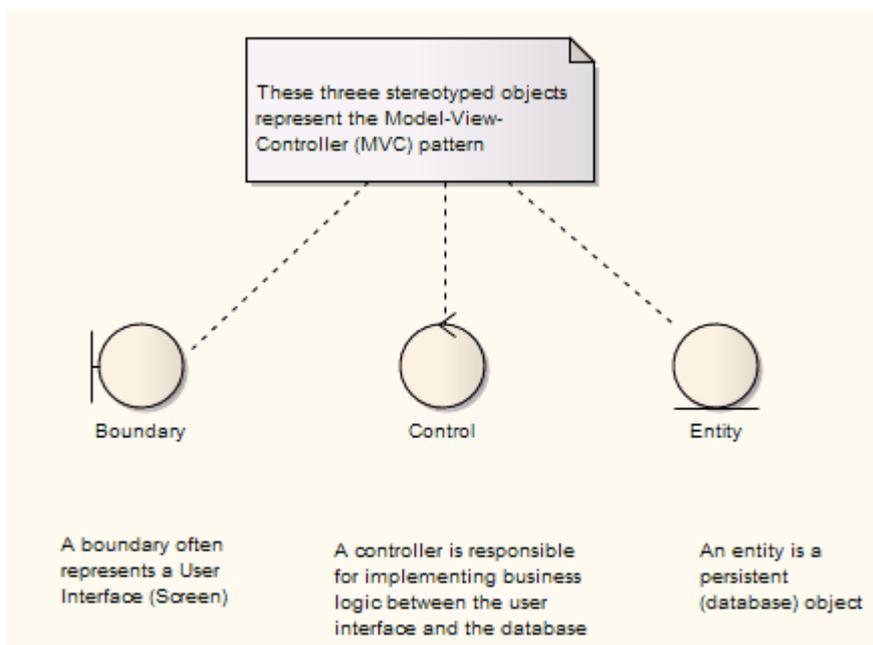
UML supports a large number of *stereotypes*, which are an inbuilt mechanism for logically extending or altering the meaning, display and syntax of a model element. Different model elements have different standard stereotypes associated with them.

For further definition of stereotypes, see the OMG UML specification (*UML Superstructure Specification, v2.1.1, section 18.3.8, pp. 667-672*).

A stereotype is generally displayed as in the example below (where «myStereotype2» is the stereotype).



In some cases the stereotype causes the element to be drawn differently, as below:



New, or customized, stereotypes can be created. Stereotypes can also be associated with new shapes, using either metafiles (image files) and colors or Shape Scripts, to apply non-UML shapes to elements and connectors. For further information on customizing stereotypes and applying Shape Scripts, see the MDG Technology SDK topic.

Learn More:

- [Apply Stereotypes](#) ^[1017]
- [Stereotype Selector](#) ^[1017]
- [Stereotype Visibility](#) ^[1018]
- [Standard Stereotypes](#) ^[1019]
- [Stereotypes with Alternative Images](#) ^[1020]

- [MDG Technology SDK](#)^[1038]
- [Customizing Stereotypes](#)^[1039]
- [Shape Scripts](#)^[1039]

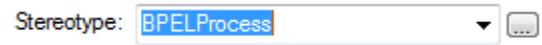
6.6.1 Apply Stereotypes

Enterprise Architect enables you to apply one or more stereotypes to any UML construct, including:

- Elements (such as Classes and Objects)
- Relationships (such as Dependencies and Associations)
- Association Ends
- Attributes and Operations
- Operation Parameters

How to:

To apply a stereotype to any UML construct, using the Properties *dialog*, select any one of the following steps

Step	Action
1	In the Stereotype field, type the stereotype(s) to apply as a comma-separated list
2	Click on the drop-down arrow and select the required stereotype from the list
3	Click on the (...) button to use the Stereotype Selector dialog 

To apply a stereotype to an element using the Properties *window*, select any of the following steps

Step	Action
1	In the Stereotype field, type the stereotype(s) to apply as a comma-separated list
2	Click on the drop-down arrow and select the required stereotype from the list
3	Select the browse other stereotypes... option in the drop-down list to use the Stereotype Selector dialog

6.6.2 Stereotype Selector

The Stereotype Selector dialog enables you to apply one or more stereotypes to a UML construct, from multiple stereotype sources such as Profiles or the Custom Stereotypes list. The appearance of the stereotype is influenced by the stereotype visibility settings on the Diagram Properties dialog.

How to:

To select stereotypes to apply/remove, follow the steps below:

Step	Action
1	On the element or connector Properties dialog, click on the (...) button near the Stereotype field

Step	Action
	The Stereotype for:<object type> dialog displays
2	Click on the Profile drop-down arrow and choose the required stereotype source
3	In the Stereotypes list, enable or disable the required stereotype by selecting or deselecting the checkbox against it
4	Click on the OK button to apply the selection

You can also define a new stereotype to apply to the required construct by clicking on the **New...** button and entering the name of the new stereotype when prompted

Learn More:

- [Applying Stereotypes](#)^[1017]
- [Stereotype Visibility](#)^[1018]

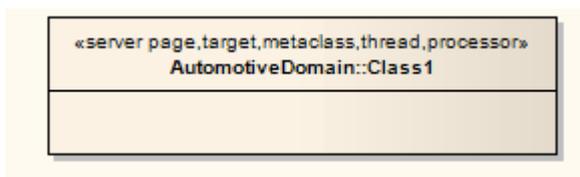
6.6.3 Stereotype Visibility

You control the visibility of applied stereotypes using three options in the diagram **Properties** dialog. Select:

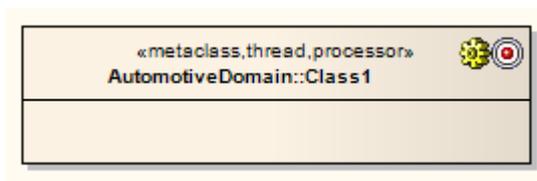
- The **Show Element Stereotypes** checkbox (on the **Elements** page) to show or hide all element stereotypes in the current diagram
- The **Show Stereotypes** checkbox (for features, on the **Features** page) to show or hide all attribute and operation stereotypes in the current diagram
- The **Use Stereotype Icons** checkbox (on the **Elements** page) to display icons, instead of strings, for those stereotypes that have icons defined.

The example below shows how a Class would appear having multiple stereotypes applied to it:

Use Stereotype Icons disabled: displays all the applied stereotypes in a comma-separated string within «guillemets».



Use Stereotype Icons enabled: displays icons for those stereotypes with icons defined. Stereotypes without icons defined are still displayed in the comma-separated string.



6.6.4 Standard Stereotypes

Below is a list of standard element stereotypes (as provided in the *EABase.eap* base model), each enclosed by guillemets (« »):

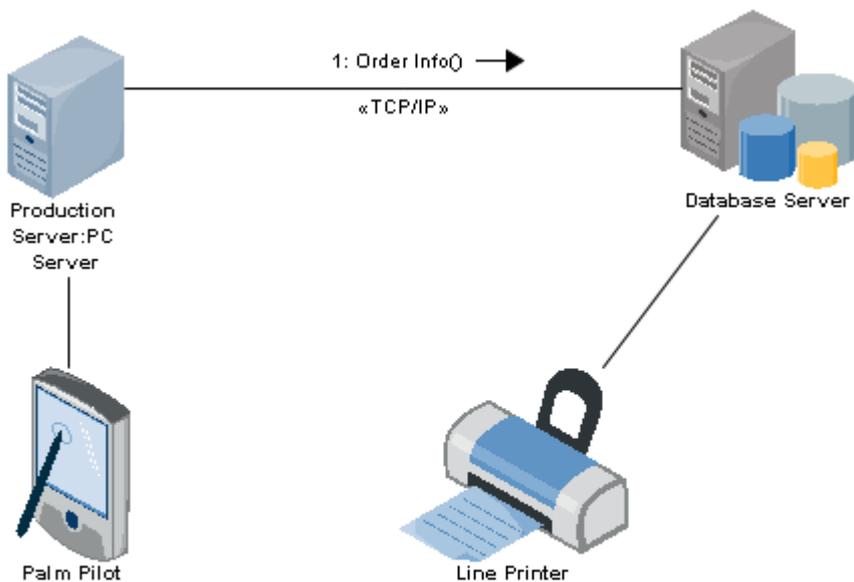
Stereotype	Base Class
«access»	Dependency
«become»	Flow
«call»	Usage
«copy»	Flow
«create»	Message
«derive»	Abstraction
«destroy»	Message
«document»	Abstraction
«executable»	Abstraction
«facade»	Package
«file»	Abstraction
«framework»	Package
«friend»	Dependency
«global»	AssociationEnd
«implementation»	Class
«implementation»	Generalization
«import»	Dependency
«instantiate»	Usage
«invariant»	Constraint
«library»	Abstraction
«local»	AssociationEnd
«metaclass»	Class
«parameter»	AssociationEnd
«postcondition»	Constraint
«powertype»	Class
«precondition»	Constraint
«process»	Classifier
«refine»	Abstraction
«requirement»	Comment

Stereotype	Base Class
«responsibility»	Comment
«self»	AssociationEnd
«send»	Usage
«stub»	Package
«table»	Abstraction
«thread»	Classifier
«trace»	Abstraction
«type»	Class
«utility»	Classifier

6.6.5 Stereotypes with Alternative Images

You can alter the appearance of elements using stereotypes. This does not apply to elements that include Lifelines, such as those in Sequence diagrams.

If the stereotype has an associated metafile, when the stereotype is applied to a Class or other element that supports alternative graphical format, Enterprise Architect then draws the alternative image instead of the standard one.



Metafiles let you specify physical diagrams in recognisable images - use the Reference/Stereotype dialog to associate metafiles with stereotypes

6.7 UML Patterns

Patterns are a group of collaborating Objects/Classes that can be abstracted from a general set of modeling scenarios. They are also known as parameterized collaborations.

Patterns are an excellent means of achieving re-use and building in robustness. As patterns are discovered in any new project, the basic pattern template from previous engagements can be re-used with the appropriate variable names modified for the current project.

Patterns generally describe how to solve an abstract problem, and it is the task of the pattern user to modify the pattern elements to meet the demands of the current engagement.

Before using a pattern it must first be created as a standard UML diagram and then saved as an XML pattern file. This XML file can then be imported as a UML resource that can be used in any model.

Sparx-Created GoF Patterns

To get you started with design patterns in Enterprise Architect, Sparx Systems provides you with an MDG technology for the patterns described in the book *Design Patterns - Elements of Reusable Object-Oriented Software* by Gamma et al., referred to as the 'Gang of Four' or GoF. These patterns are made available through a set of Toolbox pages.

The pattern elements are drawn from the EABase.eap file, through the **Resources** window hierarchy. Therefore, if you are developing your model in a DBMS repository (or you inadvertently delete the GoF patterns from your .eap file) you can download them as a 'zip' file from www.sparxsystems.com/uml_patterns.html.

Because the patterns are drawn from the **Resources** window, if you delete a pattern in the **Resources** window the equivalent **Toolbox** item cannot work. Therefore, if you cannot drop a pattern element from the **Toolbox**, check that it is still available in the **Resources** window.

See Also

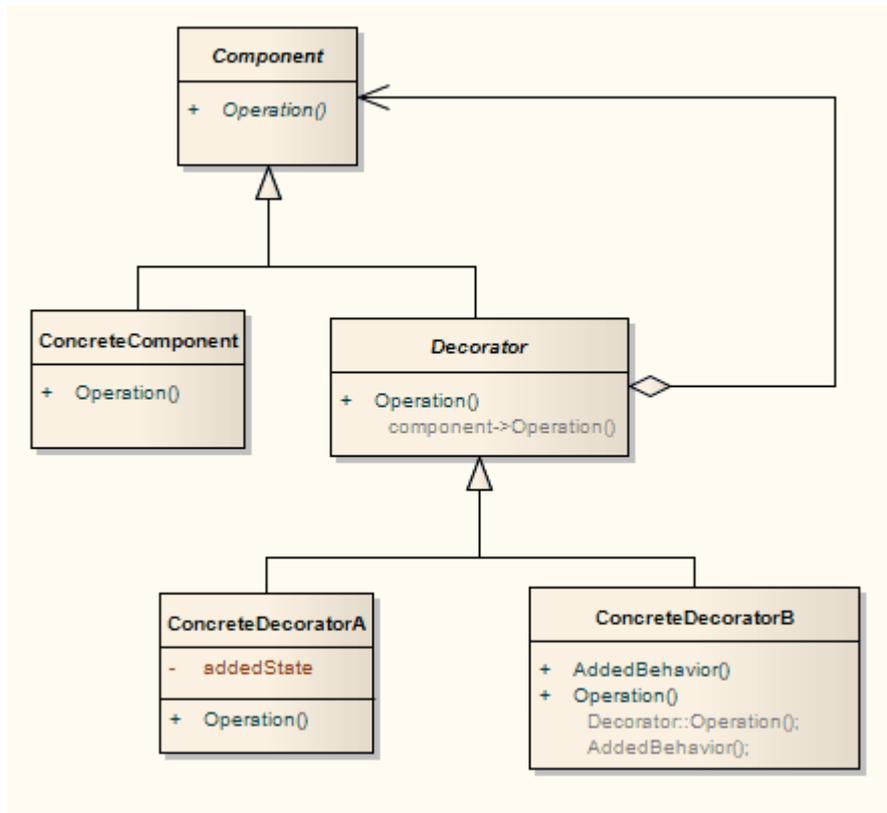
- [Create a Pattern](#)^[1022]
- [Import a Pattern](#)^[1023]
- [Use a Pattern](#)^[1024]

Learn More

- [GoF Patterns](#)^[1545]

6.7.1 Create a Pattern

To create a Pattern you first must model the Pattern as a standard UML diagram within Enterprise Architect. The following diagram was created from an example in the GoF book *Design Patterns - Elements of Reusable Object-Oriented Software* by Gamma et al.



How to:

To save a diagram as a Pattern, follow the steps below:

Step	Action	See Also
1	Select the Diagram Advanced Save UML Pattern menu option The Save Diagram as UML Pattern dialog displays	
2	In the Pattern Name field, type the Pattern name	
3	In the Filename field, type a directory path and .XML filename into which to save the Pattern	
4	In the Category field, type the Category under which the Pattern should be listed in UML Patterns (required)	
5	In the Version field, type the Pattern version number, and in the Notes field type any notes on the Pattern	
6	Select the actions for the elements that are contained in the Pattern by selecting the appropriate checkboxes; these actions are performed when the Pattern is used	Use a Pattern 1024

Step	Action	See Also
	<p>The available actions are:</p> <ul style="list-style-type: none"> • Create: Creates the Pattern element directly without modification • Merge: Merges the Pattern element with an existing element, enabling the existing element to take on the role of the selected Pattern element • Instance: Creates the Pattern element as an instance of an existing element • Type: Creates the Pattern element types as an existing element <p>If your Pattern includes an Object element, you would use Instance to set the classifier of the Object to one of the Classes in the diagram onto which you are dropping the Pattern</p> <p>If your Pattern includes a Property (Port or Part) you would use Type to set the type of the Property to one of the Classes in the diagram onto which you are dropping the Pattern</p>	
7	<p>To change the name of one of the elements, double-click on the element to display the Edit dialog</p> <p>From this dialog you can also add comments detailing the element's purpose</p>	
8	<p>Click on the OK button twice to save the Pattern</p> <p>Once saved you can load it into Enterprise Architect as a Pattern in the Resources window</p>	Import a Pattern <small>[1023]</small>

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to save a diagram as a Pattern
- If your source diagram contains information flows, the Information Items Conveyed and Information Flows Realized data is not copied into the Pattern

6.7.2 Import a Pattern

Before using a previously created Pattern file in a UML model, you must first import it into the model; it is then available from the Resources window and optionally from the Toolbox

How to:

To import a UML Pattern you have previously saved, follow the steps below:

Step	Action
1	Select the Resources window
2	<p>Right-click on the <i>UML Patterns</i> node</p> <p>The context menu displays</p>
3	<p>Select the Import UML Pattern menu option</p> <p>The Select UML Pattern Import Filename dialog displays</p>
4	Locate the XML file to import
5	Click on the Open button to import the Pattern

Step	Action
	The imported Pattern is placed in the appropriate category as defined in the XML file; if the category does not already exist under UML Patterns, a new one is created

Gang of Four patterns are integrated with Enterprise Architect in the *EABase.eap* file; however, if you create your model in a DBMS repository (or you inadvertently delete the patterns from your model .eap file) you can use the above procedure to download examples of the Gang of Four patterns from the GoF Patterns zip file on the Sparx Systems website.

Learn More:

- [Create a Pattern](#)^[1022]
- [GoF Patterns zip file](#)

6.7.3 Use a Pattern

Using a Pattern enables you to rapidly create template solutions for code structures that perform the same type of task in other situations, and to use items defined in the Pattern with the UML model

How to:

To use a Pattern that you have previously imported into the model, follow the steps below:

Step	Action	See Also
1	Open the diagram into which to add the UML Pattern	
2	Select the Resources window	
3	Expand the <i>UML Pattern</i> folder and find the Pattern to add	
4	Either: <ul style="list-style-type: none"> • Right-click on the Pattern and select the Add Pattern to Diagram context menu option or • Drag and drop the Pattern from the Resources window onto the diagram You can also view the Pattern details in read-only mode by selecting the View Pattern Details context menu option	Add Pattern Dialog ^[1025]
5	Once the appropriate selections have been made, click on the OK button to import the Pattern into the model, recreating the original diagram with new GUIDs	

To change the default of the Pattern element, follow the steps below:

Step	Action	See Also
1	From the Add Pattern dialog select the individual element in the Pattern Element panel	
2	Click on the (...) button to display the Edit dialog The specific method for changing the element name is dependant upon the entry	

Step	Action	See Also
	in the Action column of the Pattern Elements panel	
3	If the Action entry is Create , then in the Default field in the Edit dialog delete the existing value and type your own, user-defined value Click on the OK button The element default is updated on the Add Pattern dialog	
4	If the Action entry for the element is Merge , in the Edit dialog click on the (...) button to browse to an existing element classifier The Select <Item> dialog displays	
5	Locate and select an existing element classifier You can restrict the number of choices by selecting the elements from a specific namespace; to do this, click on the In Namespace drop-down arrow and select a namespace	Using Classifiers ^[706]

Learn More:

- [Import a Pattern](#)^[1023]

6.7.4 Add Pattern Dialog

The Add Pattern dialog displays when you are using or editing a Pattern element.

Panel	Action
Preview	Display a preview of the Pattern; click on the Preview link to open a view of the Pattern and drag the sides into as large a picture as you require
Pattern Elements	Access the individual elements contained in the Pattern From here you can: <ul style="list-style-type: none"> • Select the action for the individual element (<i>Create</i>, <i>Merge</i>, <i>Instance</i> or <i>Type</i>, as applicable for each element) by clicking on the drop-down arrow, or • Modify the default of the Pattern element or - for a merged element - choose the namespace, by clicking on the (...) button on the right of the Default entry
Element Notes	Display the comments that describe the element in the Pattern Highlight an element in the Pattern Elements panel to view the notes

6.8 UML Profiles

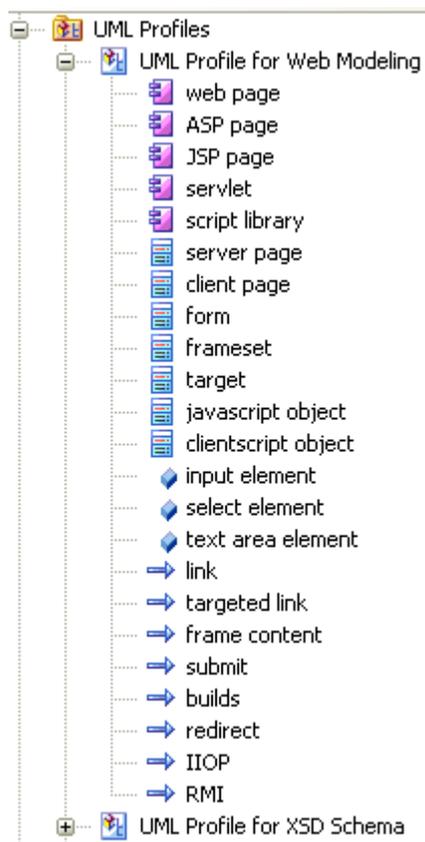
UML Profiles provide a means of extending UML, which enables you to build UML models in particular domains. They are based on additional stereotypes and Tagged Values that are applied to elements, attributes, methods, connectors, connector ends and so on. A Profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain. For example, the UML Profile for XML describes a set of extensions to basic UML model elements to enable accurate modeling of XSD Schemas (see *Modeling XML Applications with UML*, David Carlson, p. 310).

Enterprise Architect has a generic UML Profile mechanism for loading and working with different Profiles. UML Profiles for Enterprise Architect are specified in XML files, with a specific format. You can reference these XML files in Enterprise Architect as part of an MDG Technology or import them through the Resources window. Once imported, you can drag and drop Profile elements onto the current diagram. Enterprise Architect attaches the stereotype, Tagged Values and default values, notes and metafile if one is specified, to the new element. You can also drag and drop attributes and operations onto existing Classes and have them immediately added with the specified stereotype and values.

The imported Profile also automatically generates a page of elements and relationships in the Toolbox.

Profiles in the Resources Window:

The Resources window contains a tree structure with entries for items such as MDG Technologies, Documents, Stylesheets, Matrix profiles and UML Profiles. The *UML Profiles* node initially contains no entries; to be able to use Profiles you must import them into Enterprise Architect from supplied XML files.



Items in the Profile represent stereotypes. These can be applied to UML elements in the following ways:

- Stereotypes that apply to elements such as *Classes* and *interfaces* can be dragged directly from the Resources window to the current diagram, automatically creating a stereotyped element; alternatively, they can be dragged onto existing elements, automatically applying them to the element

- Stereotypes that apply to *attributes* can be drag-and-dropped onto a host element (such as a Class); a stereotyped attribute is automatically added to the element's feature list
- Stereotypes that apply to *operations* are like those that apply to attributes; drag-and-drop onto a host element to add the stereotyped operation
- Stereotypes that apply to *connectors* such as *associations*, *generalizations*, *messages* and *dependencies* are added by selecting them in the Project Browser, then clicking on the start element in a diagram and dragging to the end element (in the same manner as adding normal connectors); a stereotyped connector is added
- Stereotypes that apply to *association ends* can be added by dragging the connector end element over the end of an Association in the diagram

To get you started, some Profiles are supplied on the Sparx Systems website. You can download these and import them into Enterprise Architect. Over time Sparx Systems intend to expand the range of Profiles, the content of each Profile and the degree of customization possible in each Profile.

You can also create your own Profiles to describe modeling scenarios specific to your development environment.

Learn More:

- [MDG Technology](#)^[1037]
- [MDG Technology SDK](#)^[1038]
- [Import a UML Profile](#)^[1027]
- [Add Profile Objects to a Diagram](#)^[1028]
- [Tagged Values in Profiles](#)^[1028]
- [Synchronize Tagged Values and Constraints](#)^[1029]
- [Profiles webpage](#)

6.8.1 Import a UML Profile

To import a Profile you must have a suitable Profile XML file, such as the Profiles supplied on the Sparx Systems website. If the Profile includes references to any metafiles, they should be in the same directory as the Profile XML file.

How to:

To import a Profile, follow the steps below:

Step	Action
1	Open the Resources window
2	Right-click on the <i>UML Profiles</i> tree node and select the Import Profile context menu option The Import UML Profile dialog displays
3	Locate the XML Profile file to import using the (...) (Browse) button
4	Set the required import option checkboxes for all stereotypes defined in the Profile; you can select: <ul style="list-style-type: none"> • Element Size - to import the element size attributes • Color and Appearance - to import the color (background, border and font) and appearance (border thickness) attributes • Alternate Image - to import the metafile image • Code Templates - to import the code templates if they exist • Overwrite Existing Templates - to overwrite any existing code templates defined in the current project

Step	Action
5	Click on the Import button The Profile is added to the <i>UML Profiles</i> folder

Notes:

- If the Profile already exists, Enterprise Architect prompts you to overwrite the existing version and import the new one (or cancel); once the import is complete, the Profile is ready to use
- This topic describes importing a stand-alone Profile; you can also embed Profiles in an MDG Technology and import the Technology file into the Enterprise Architect installation directory

Learn More

- [Profiles webpage](#)
- [Add Profile Objects and Features to a Diagram](#) ^[1028]
- [Creating MDG Technologies](#) ^[1064]
- [Deploying MDG Technologies](#) ^[1089]

6.8.2 Add Profile Objects to a Diagram

After you have imported a Profile into the Resources window, the profiled objects (elements and connectors) and features (attributes and operations) are available from the expanded Profile folder in the window and from the appropriate pages of the Toolbox (click on the **More tools** option at the top of the Toolbox).

Similarly, when you import an MDG Technology, it adds the appropriate pages of profiled elements and connectors to the Toolbox.

Action	Description
Add a Profile-based element to a diagram	Click on the element in the Toolbox page or the Resources window, and drag it onto the diagram
Add a Profile-based connector to a diagram	Click on the connector in the Toolbox page or Resources window, then click on the source element in the diagram and drag it to the target You can also drag the connector from the Resources window to the source, which automatically displays a list box of target elements; select a target from the list to create the connector to that target
Add a profile-based attribute or operation to a diagram	Click on the attribute or operation in either the Toolbox page or the Resources window, and drag it onto the host element on the diagram The system prompts you to enter a name for the feature

Learn More:

- [Toolbox](#) ^[546]
- [MDG Technology](#) ^[1031]

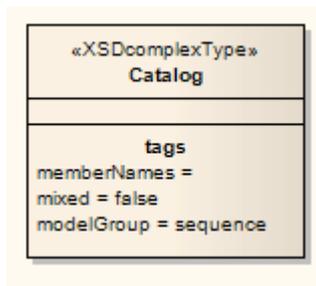
6.8.3 Tagged Values in Profiles

Stereotypes within a UML Profile can have one or more associated Tagged Values. When you create an element based on a UML Profile Stereotype by dragging from the Resources window to a diagram, any associated Tagged Values are added to the element as well. Tagged Values and Profiles are an excellent way to extend the use of Enterprise Architect and the power of UML modeling.

For example, in the UML Profile for XSD, there is an *XSDComplexType* stereotype, which has the following Tagged Value declaration:

```
<TaggedValues>
  <Tag name="mixed" description="Determines whether this element can
  contain mixed element and character content. See the W3C XML Schema
  recommendation" />
  <Tag name="modelGroup" description="Overrides the package-level
  default model group" values="all | sequence | choice"
  default="sequence" />
  <Tag name="memberNames" description="Overrides the package-level
  default for naming complexType definitions" />
</TaggedValues>
```

When you create an element from the *XSDComplexType* stereotype (by dragging from the Profile Elements page of the Toolbox onto a diagram), the Tagged Values are added automatically.



Tagged Values that have default values are automatically set and displayed in the element *tags* section, if applicable. When you select the element, the Tagged Values window displays all the associated tags, including ones that have no value set. Also note that Tagged Values in the Profile that have a *Values* section (for example, *values="element | attribute | both" default="both"*) enable you to select the non-default values from a drop-down list. Where no *Value* list exists, the tag accepts free text.

Learn More:

- [Predefined Structured Types](#) ^[108]

6.8.4 Synchronize Tagged Values and Constraints

When you create an element, attribute, operation or link from a UML Profile item, you add the Tagged Values and constraints from the Profile. Over time you might modify the constraints or the notes and tags of the Tagged Values of a particular profiled item, so the items already created might be missing additional Tagged Value tags and notes, or constraints.

Similarly, you might have manually set the stereotype on a set of elements and now want them to receive the Tagged Values and constraints associated with that stereotype.

To make sure you have all the related Tagged Values and constraints, use the Synchronize Stereotype function. This operates on both:

- Profiles created in an MDG Technology File and not held in the Resources window - either an in-house customized Add-In, an imported external technology or an integrated technology provided with Enterprise Architect such as BPMN 1.1
- Profiles imported to the Resources window

When a Profile or MDG Technology file is deployed in Enterprise Architect, the Profile can be accessed through a set of Toolbox pages for that Profile or Technology. The profiled elements in these Toolbox pages automatically trigger an additional context menu option, **Synchronize Stereotype**.

How to:

To synchronize elements using the Profile or Technology pages of the *Toolbox*, follow the steps below:

Step	Action
1	Ensure that the Toolbox displays pages containing the stereotyped Profile elements
2	Right-click on the element profile in the Toolbox (for example, the BPMN 1.1 <i>Activity</i> element) The Toolbox context menu displays
3	Click on the Synchronize Stereotype menu option The Synch Profiled Elements dialog displays
4	Click on the OK button to proceed The Actions list is populated with the items that have been modified and the changes that were made

Alternatively:

You can quickly synchronize the tags and constraints of a *single* element in a diagram by dragging the updated profile element from the Toolbox page onto the element in the diagram, then selecting the **Apply «stereotype»** context menu option.

Notes:

- You can review any changes by displaying the element Properties dialog and by opening the Tagged Values window and clicking on an appropriate profiled element

6.9 MDG Technologies

The Model Driven Generation (MDG) Technologies enable you to access and use resources pertaining to a specific technology in Enterprise Architect. You have various options for bringing MDG Technologies into use with Enterprise Architect:

Source of Technology	See Also
Core technologies - Enterprise Architect consists of a: <ul style="list-style-type: none"> • Basic UML 2 technology as an implementation of UML 2.3 structural and behavioral modeling, and • Core Extensions technology that applies profiles and stereotypes to provide extended modeling of aspects such as Requirements, User interface and Data Modeling 	
Sparx Systems already provides some added technologies in the Enterprise Architect Install directory, <i>MDGTechnologies</i> subfolder	MDG Technologies ^[1038]
You can import technologies into the APPDATA folder (%APPDATA%\Sparx Systems\EAMDGTechnologies)	Import MDG Technologies ^[1035]
You can transfer technologies into the MDGTechnologies subfolder; these technologies are available when you restart Enterprise Architect (on Vista/Windows 7 systems you might have to increase your access permissions to do this)	
You can access and activate MDG Technologies in remote system folders or web sites, from Enterprise Architect	Access Remote Technologies ^[1034]
Technology Developers can create new MDG Technologies and deploy them to the project team either through the MDGTechnologies subfolder or from any other remote folder or website	Create New Technologies ^[1038]

To see which technologies are available within Enterprise Architect, and activate the ones you require, use the MDG Technologies dialog.

Having made the MDG Technologies available to Enterprise Architect, you can manage their availability to users and you can work with them.

You also have the facility to turn off or disable the Enterprise Architect Basic UML 2 and Core Extensions Toolbox pages and facilities, so that you can apply the Enterprise Architect facilities and features exclusively to one or more selected MDG Technologies.

Learn More:

- [Work with MDG Technologies](#) ^[1037]
- [Manage MDG Technologies](#) ^[1035]

6.9.1 Work with MDG Technologies

Any MDG Technology listed on the MDG Technologies dialog can be enabled, which makes their interface profiles and Toolbox pages available for your use.

Topic	Detail	See Also
MDG Technology Toolbox Pages	When you enable an MDG Technology, any Technology-specific diagram types are added to the New Diagram dialog lists, and the Technology's Toolbox pages are added	Create New Diagrams ^[568]

Topic	Detail	See Also
	<p>to those available through the More tools menus in the Toolbox</p> <p>If you set the MDG Technology to <i>Active</i>, its Toolbox pages override any parallel Enterprise Architect Toolbox pages; for example, the ICONIX <i>Class</i> pages would override the Enterprise Architect <i>Class</i> pages</p> <p>You create Technology-specific diagrams and populate them with elements and connectors in the same way as for standard Enterprise Architect diagrams</p>	
The Resources Window	<p>A method of importing MDG Technologies into the Resources window is available (click on the <i>MDG Technologies</i> folder and select the Import Technology option) but this method is Not Recommended</p> <p>If you use this method, the MDG Technology is available only in the model it was imported into, and the Toolbox pages, Learning Center, Project Browser icons and model templates are not available</p> <p>It is recommended that you instead create or download new technologies into the %APPDATA% folder or the Enterprise Architect installation directory, or use new technologies from remote file locations and web sites</p> <p>However, you might previously have imported technologies into the Resources window, and these are still available until you specifically delete them (right-click on the technology and select the Delete Technology context menu option)</p>	<p>Import MDG Technologies <small>[1033]</small></p> <p>Access Remote MDG Technologies <small>[1034]</small></p>

The following information is provided to support implementations where technologies have been previously imported via the Resources window.

The Resources window (**View | More Project Tools | Project Resources**) displays a tree structure containing nodes such as imported MDG Technologies, Templates, Documents, Stylesheets, Matrix profiles and UML Profiles.

MDG Technologies can bundle the functionality provided by UML Profiles, UML Patterns, Code Templates and Model Types.

Profiles contained in MDG Technologies are applied to:

- Elements such as Classes and Interfaces, which are dragged directly from the Toolbox or the Resources window to the current diagram
- Attributes, which are dragged over a host element (such as a Class) to be automatically added to the element feature list
- Operations which, like Attributes, are dragged over a host element to add the operation
- Connectors such as Association, Generalization, and Dependency, which are added by selecting them in the Toolbox or Resources window, then clicking on the source element in a diagram and dragging to the target element (in the same way as adding normal connectors); the connector is added with the new stereotype and Tagged Value information
- Association Ends, which are added by dragging the connector end element over the end of an Association in the diagram

Patterns contained in MDG Technologies are used to:

- Enable reuse in a model
- Build in robustness

Code Templates are used to:

- Specify the transformation from UML elements into various parts of a given programming language

Model Types are used to:

- Define the data types for the model

6.9.2 Manage MDG Technologies

You use the MDG Technologies dialog to manage the MDG Technologies available and accessible to Enterprise Architect users.

Access: [Settings](#) | [MDG Technologies](#)

Topic	Detail	See Also
MDG Technologies Dialog	The MDG Technologies dialog lists the technologies held in the APPDATA folder (marked by the text <i>Location: Technology.xml</i>) and Enterprise Architect Install directory, in alphabetical order	Import MDG Technologies 
Enable and Disable MDG Technologies	<p>All MDG Technologies listed can be made available (enabled) or removed from use (disabled); to enable or disable a Technology, click on its Enabled checkbox</p> <p>When an MDG Technology is enabled, three things happen:</p> <ul style="list-style-type: none"> • The MDG Technology is added to the list of available options in the profile field of the Default Tools toolbar, so that you can apply the interface profiles of the MDG Technology • At least one set of Toolbox pages for the MDG Technology is automatically added to the Toolbox; you can access the added Toolbox pages through the More Tools menu • Any MDG Technology-specific diagram templates are added to the New Diagram dialog for selection; when selected, these display the diagram-specific Toolbox pages <p>You can quickly enable or disable all the listed MDG Technologies by clicking on the All or None buttons; however, if you click on the None button, you should scroll to the top of the list and select the Basic UML 2 Technology and Core Extensions checkboxes to re-enable the UML and Extended Toolbox pages and diagram types</p>	
Set as Default	<p>You can make an MDG Technology the default interface to Enterprise Architect. Depending on the MDG Technology selected, this can change the way Enterprise Architect windows are displayed and override the Toolbox pages with pages specific to that Technology.</p> <p>To set an MDG Technology as the default interface, click on it in the Technology panel and click on the Set Active button.</p> <p>This displays an asterisk against the MDG Technology name in the Technology panel, and selects the MDG Technology in the profile field of the Default Tools toolbar. If</p>	

Topic	Detail	See Also
	the MDG Technology has not been enabled, this also enables it.	
Basic UML 2 Technology	You can also enable one or more of the MDG Technologies (and likely make one of them the default) and then <i>deselect</i> the Basic UML 2 Technology checkbox (and, if necessary, the Core Extensions checkbox), to work exclusively in the selected technologies. The UML Toolbox pages, diagram types and quicklinks are excluded from the Toolbox, More tools menu, diagrams and New Diagram dialog in the user interface.	
MDG Technologies Outside Enterprise Architect	The MDG Technologies dialog lists technologies that have been loaded into the APPDATA folder and Enterprise Architect install directory. You can also add MDG Technologies in folders and websites remote from Enterprise Architect. To do this, click on the Advanced button.	Access Remote MDG Technologies <small>[1034]</small>

6.9.3 Access Remote MDG Technologies

You can access MDG Technologies in folders and websites remote from Enterprise Architect.

If you have not already identified the location of the MDG Technology, you must first do this.

Later, if you have no further use for the MDG Technology, you can remove it from the list of identified MDG Technologies.

If you add or remove remote MDG Technologies, you must *restart* Enterprise Architect to show them on or remove them from the list on the MDG Technologies dialog, and to use them.

Access: **Settings | MDG Technologies**

How to:

To specify the location of a remote MDG technology, follow the steps below:

Step	Action	See Also
1	Select the MDG Technologies menu option The MDG Technologies dialog displays	Manage MDG Technologies <small>[1033]</small>
2	Click on the Advanced button The MDG Technologies - Advanced dialog displays	
3	Click on the Add button A short context menu displays, offering the options: <ul style="list-style-type: none"> • Add Path • Add URL 	
4	To specify an MDG Technology <i>in a directory folder</i> , select the Add Path option	

Step	Action	See Also
	The Browse for Folder dialog displays Browse for the MDG Technology folder, click on it, and click on the OK button; go to step 6	
5	To specify an MDG Technology <i>on a web site</i> , select the Add URL option The Input dialog displays In the Enter Value field, type or copy-and-paste the MDG Technology URL and click on the OK button	
6	The folder path or URL for the MDG Technology displays in the Path panel	

To remove an MDG Technology listed in the MDG Technologies - Advanced dialog, click on the folder path or URL and click on the **Remove** button; the path or URL is deleted

6.9.4 Import MDG Technologies to APPDATA

The preferred method of making an MDG Technology available to your project team members is to import the technology into the %APPDATA%\Sparx Systems\EA\MDGTechnologies folder. To import an MDG Technology you must have a suitable MDG Technology XML file. If the MDG Technology includes references to any metafiles, they should be in the same directory as the MDG Technology XML file.

On startup, Enterprise Architect scans the APPDATA folder (as well as the Enterprise Architect Install directory *MDGTechnologies* subfolder) for technology files, to make them available through the MDG Technologies dialog. Technologies imported to the APPDATA folder are indicated by the text *Location: Technology.xml*.

Access: [Tools | MDG Technology Import](#)

How to:

To import a technology, follow the steps below:

Step	Action
1	Select the MDG Technology Import menu option The Copy Technology to Application Data dialog displays
2	In the Filename field, type the path and filename of the MDG Technology file to import, or browse for it using the (...) button When you enter the filename, the MDG Technology name and version display in the Technology and Version fields
3	Click on the OK button <ul style="list-style-type: none"> • If the APPDATA folder does not yet exist, Enterprise Architect creates it • If the MDG Technology already exists, Enterprise Architect displays a prompt to overwrite the existing version and import the new one Once the import is complete, you must restart Enterprise Architect; the MDG

Step	Action
	Technology is then listed in the MDG Technologies dialog

Notes:

- This feature is not available in the Desktop edition of Enterprise Architect

Learn More:

- [Manage MDG Technologies](#)^[1035]

6.9.5 Extensions - MDG Technologies

Enterprise Architect is the core for a range of Model Driven Generation (MDG) extensions to its modeling capabilities, using more specialized, niche frameworks and profiles.

Topic	See Also
A number of technologies are already integrated with the Enterprise Architect installer	ArchiMate ^[1272] BPEL ^[1235] BPMN ^[1219] Data Flow Diagrams ^[1184] Eriksson-Penker Extensions ^[1273] ICONIX ^[1542] Mind Mapping ^[1182] SoaML ^[1624] SOMF ^[1628] Strategic Modeling Systems Modeling Language (SysML) ^[1552]
Enterprise Architect provides support for: <ul style="list-style-type: none"> • Downloading MDG Technologies from external system files or websites, or • Creating your own easily with the Enterprise Architect MDG Technology Wizard 	Access Remote MDG Technologies ^[1034] MDG Technology SDK ^[1038]
Sparx Systems also market a number of MDG products, as follows: <ul style="list-style-type: none"> • MDG Technology For: <ul style="list-style-type: none"> • Zachman Framework • The Open Group Architecture Framework (TOGAF) • Unified Profile for DoDAF and MODAF (UPDM) • Data Distribution Service (DDS) • Python (for Enterprise Architect versions 4.5 to 5.0, integrated in later versions) (* free product! *) • CORBA (* free product! *) • Java Beans (* free product! *) • Testing (* free product! *) • MDG Integration For: <ul style="list-style-type: none"> • Eclipse 3.3 	

Topic	See Also
<ul style="list-style-type: none"> • Visual Studio 2005 and 2008 • Siemens PLM Teamcenter Systems Engineering (TcSE) • MDG Link For <ul style="list-style-type: none"> • Eclipse • Visual Studio.NET • Microsoft Visio (* free product! *) • Telelogic DOORS <p>Over time, this list is being extended to include further products</p>	
<p>Sparx Systems provide different editions of Enterprise Architect tailored for systems engineering and business engineering, or both together</p> <p>These editions incorporate several of the above MDG Technologies and other Add-Ins</p>	Enterprise Architect Editions ^[13]
<p>Product Information: For the latest list of available Add-Ins and an introduction to each product, including details of pricing, purchasing and download options, see the Sparx Systems website</p> <p>When you purchase one of the Add-Ins, you receive one or more license keys and instructions on obtaining, installing and registering the product</p> <p>The information page for most products provides a link to download the product User Guide in .pdf format</p> <p>The product User Guide can also be displayed as a .chm file online within the product itself; to access this online help in Enterprise Architect, select the Add-Ins <productname> Help menu option</p>	Products Page on the Sparx Systems website

Learn More:

- [Access Remote MDG Technologies](#) ^[1034]
- [Creating MDG Technologies](#) ^[1064]

6.10 Define a Modeling Language



Enterprise Architect enables you to create models using UML. However, it also enables you to go much further, extending the scope both of your modeling and of the UML components you use, as outlined below.

Topic	Description	See Also
UML Stereotypes	Stereotypes are an inbuilt mechanism for logically extending or altering the meaning, display and syntax of a model element. Different model elements have different standard stereotypes associated with them. You can also define your own stereotypes.	UML Stereotypes ^[1016]
UML Profiles	UML Profiles are a means of extending UML, which enables you to build models in particular domains. A Profile is a collection of additional stereotypes and Tagged Values applied to elements, attributes, methods and connectors, which together describe some particular modeling problem and facilitate modeling constructs in that domain.	UML Profiles ^[1026]
UML Patterns	Patterns are groups of collaborating Objects/Classes that can be abstracted from a general set of modeling scenarios (that is, parameterized collaborations). They generally describe how to solve an abstract problem, and are an excellent means of achieving re-use and building in robustness.	UML Patterns ^[1027]
MDG Technologies	The Model Driven Generation (MDG) Technologies enable you to access and use the resources of a specific technology within Enterprise Architect. Interfaces to some technologies, such as BPMN and ICONIX, are integrated with Enterprise Architect, whilst interfaces to others such as Eclipse and Visual Studio can be added separately. You can also link to technologies that you have created yourself.	MDG Technologies ^[1031]

6.10.1 MDG Technology SDK

Introduction:

In describing aspects of developing technologies to use in conjunction with Enterprise Architect, it is expected that you are familiar with the concepts introduced in the main body of the *Enterprise Architect User Guide*. Wherever appropriate, cross-references to these concepts are provided in the text.

Contents:

- [Developing Profiles](#) ^[1039] (incorporating [Custom Stereotypes](#) ^[1039])
- [MDG Technologies](#) ^[1063]
- [Shape Scripts](#) ^[1089]
- [Tagged Value Types](#) ^[1108]
- [Code Template Framework](#) ^[1114]

6.10.1.1 Developing Profiles

Introduction:

Profiles provide a means of extending the UML, which enables you to build models in particular domains. They are based on additional stereotypes and Tagged Values that are applied to UML elements, connectors and their components. A Profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain. UML Profiles for Enterprise Architect are specified in XML files, with a specific format. These XML files can be imported into Enterprise Architect through the Resources window.

The imported Profile also automatically generates a page of elements and relationships in the Toolbox.

The Resources window contains a tree structure with entries for items such as MDG Technologies, Documents, Stylesheets, Matrix profiles and UML Profiles. The *UML Profiles* node initially contains no entries; to be able to use Profiles you must import them into Enterprise Architect from supplied XML files.

Items in the Profile represent stereotypes. UML supports a large number of stereotypes, which are an inbuilt mechanism for logically extending or altering the meaning, display, appearance and syntax of a model element. Different model elements have different stereotypes associated with them.

Learn More:

- [Custom Stereotypes](#) ^[1039]
- [Create Profiles](#) ^[1041]
- [Quick Linker](#) ^[1058]
- [Customize Toolbox Profiles](#) ^[1074]
- [Create Diagram Profiles](#) ^[1080]
- [Create Learning Center Profiles](#) ^[1082]
- [Tagged Value Types](#) ^[1108]
- [UML Profiles](#) ^[1026] (for more information on the use of Profiles in Enterprise Architect)

6.10.1.1.1 Custom Stereotypes

UML supports a large number of *stereotypes*, which are an inbuilt mechanism for logically extending or altering the meaning, physical appearance and syntax of a model element. Different model elements have different stereotypes associated with them.

In Enterprise Architect you can create new stereotypes with their own custom appearance. The stereotypes can be altered to make use of metafiles (image files) and customized colors, or you can make use of an Enterprise Architect Shape Script to make new element shapes to determine the shape and dimensions of the element.

Access: **Settings | UML Types > Stereotypes**

How to:

To add your own custom stereotypes, follow the steps below:

Step	Action
1	On the Stereotypes tab, type or select a Stereotype name
2	Select a Base Class from the drop-down list

Step	Action
3	Associate a Metafile with this stereotype - click on the Metafile radio button and the Assign button, and locate the required .emf or .wmf file
4	Enter optional Notes and select Default Colors for this stereotype
5	Click on the Save button to save the stereotype

The table below describes the functions of the Stereotypes tab.

Field	Usage	See also
Stereotype	Specify the name of the stereotype	
Group name	Enable grouping of stereotype features by a plural name, for attributes and operations, which is shown on diagrams in the attribute and operations compartments	
Base Class	Enable the stereotyped element to inherit the base characteristics from a pre-existing element type	
Notes	Type any notes concerning the stereotype (not the elements to which the stereotype is to be applied)	
Override Appearance		
None	Switch to the default element appearance	
Metafile	Enable an image file to be used for the appearance of the stereotype	
Shape Script	Specify custom shapes for the stereotype using the Enterprise Architect Shape Scripting language	Shape Scripts <small>1089</small>
Assign	Add the associated metafile or Shape Script from the stereotyped element	
Remove	Remove the associated metafile or Shape Script from the stereotyped element	
Default Colors		
Fill	Set the default background color of the element	
Border	Control the border color	
Font	Control the color of the stereotype font	
Reset	Reset the appearance of the element to the default element appearance	

Notes:

- You can transport custom stereotype definitions between models, using the Export Reference Data and Import Reference Data options

Learn More:

- [UML Stereotypes](#) ^[1016]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

6.10.1.1.2 Create Profiles

This topic describes how to create profiles and profile items. These creation tasks include creating the profile stereotypes, defining the metaclasses they apply to, and defining Tagged Values and constraints. The topic also describes how to export a profile for use in modeling.

How to:

To create a Profile, follow the steps below:

Step	Description	See Also
1	Create a Profile package	Create Profile ^[1041]
2	Add stereotypes and metaclasses to the profile	Add Stereotypes and Metaclasses ^[1042]
3	Define Tagged Values for the stereotypes	Define Stereotype Tags ^[1044]
4	Define constraints for the stereotypes	Define Stereotype Constraints ^[1046]
5	Add enumeration elements to define values for the Tagged Values for the stereotypes	Add Enumeration Elements ^[1047]
6	Add shape scripts for the stereotypes	Add Shape Scripts ^[1048]
7	Set the default appearance for the stereotypes	Set Default Appearance ^[1049]
8	Export the profile	Export Profile ^[1050]

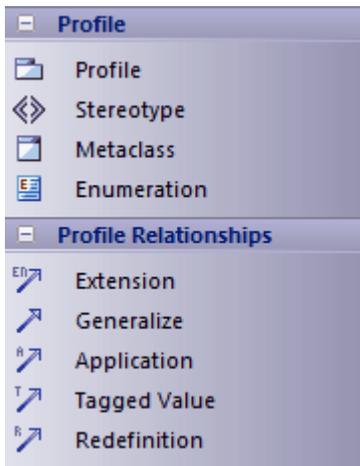
6.10.1.1.2.1 Create a Profile Package

In Enterprise Architect, you must create a Profile in a Package that has the stereotype «*profile*».

How to:

To create a Profile Package, follow the steps below:

Step	Description	See
1	Open or create a Package diagram	
2	Open the Profile page of the Toolbox (More tools Profile)	

Step	Description	See
		
3	Drag the <i>Profile</i> item onto the Class diagram The New Model Package dialog displays	
4	In the Package Name field, type a name for the Profile	
5	Select the Automatically add new diagram checkbox	
6	Click on the OK button The New Diagram dialog displays	New Diagram Dialog [568]
7	Provide the required diagram name, and select the diagram group UML Structural and diagram type Class	
8	Click on the OK button Enterprise Architect creates a package with the stereotype « <i>profile</i> » and with a child Class diagram	
9	In the Project Browser, double-click on the Profile Package on the diagram to open the child diagram You now use this child diagram to add stereotypes to the Profile	

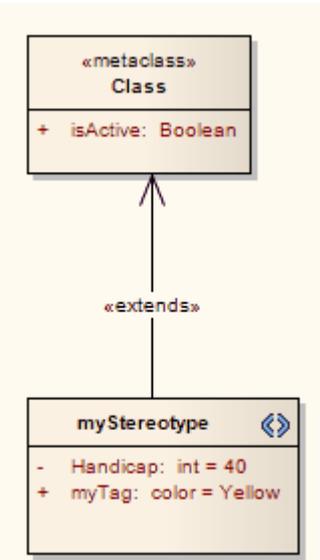
Learn More:

- [Add Stereotypes and Metaclasses](#) [1042]

6.10.1.1.2.2 Add Stereotypes and Metaclasses**How to:**

To add metaclasses and stereotypes to a Profile, for as many stereotypes and metaclasses as you require, follow the steps below:

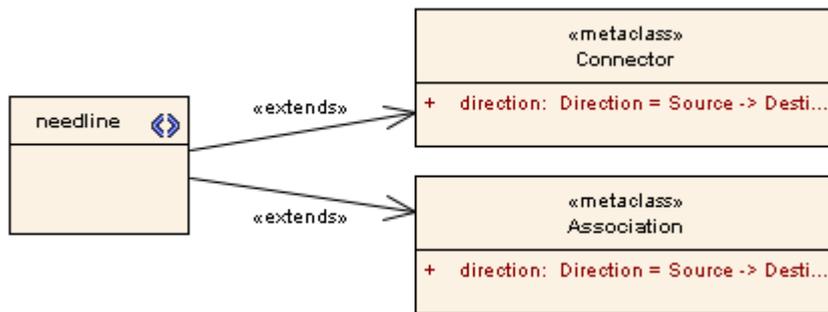
Step	Description	See Also
1	Open the child diagram of the Profile Package	
2	Drag the <i>Metaclass</i> element from the Profile page of the Toolbox onto the diagram	

Step	Description	See Also
	The Create New Metaclass dialog displays, in which you can tick multiple metaclasses for dropping onto the diagram	
3	Scroll down the Element list and select the checkbox for Class	
4	Click on the OK button The Class Properties dialog displays	
5	In the Name field type a name for the element Click on the OK button	
6	Drag a <i>Stereotype</i> element from the Toolbox onto the diagram If the Properties dialog does not display, double-click on the element on the diagram	
7	In the Name field, type a name for the stereotype	
8	Click on the OK button If it displays, Close the Generate Code dialog	
9	Click on the <i>Extension</i> relationship in the Toolbox and drag the connection from the stereotype element to the metaclass element	
10	Your diagram should now resemble the one below:  The diagram shows two class boxes. The top box is a metaclass named 'Class' with the stereotype «metaclass» and a public attribute 'isActive: Boolean'. The bottom box is a stereotype named 'myStereotype' with the stereotype «stereotype» and two attributes: a private attribute 'Handicap: int = 40' and a public attribute 'myTag: color = Yellow'. An arrow points from the 'myStereotype' box to the 'Class' box, labeled «extends».	

Notes:

- If you want to have a stereotype extending more than one metaclass, do not create two stereotype Classes with the same name

You cannot have two stereotypes with the same name in the same profile, one is discarded when you save the profile; therefore, create one stereotype Class with an Extension connector to each of several Metaclass elements, as shown below.



You can now add stereotype Tags, Constraints, Enumerations, and/or Shape Scripts to your Profile, and define the default appearance of the elements or connectors as required

Learn More:

- [Define Stereotype Tags](#) ^[1044]
- [Define Stereotype Constraints](#) ^[1046]
- [Adding Enumeration Elements](#) ^[1047]
- [Adding Shape Scripts to UML Profiles](#) ^[1048]
- [Setting Default Appearance in UML Profiles](#) ^[1049]

6.10.1.1.2.3 Define Stereotype Tagged Values

Stereotypes within a UML Profile can have one or more associated Tagged Values. When creating a UML Profile, you define these Tagged Values as attributes of the stereotyped Class.

How to:

To define Tagged Values for a stereotype, follow the steps below:

Step	Description	See Also
1	Open the Attributes dialog for the stereotyped element	
2	Click on the New button to create a new attribute	
3	In the Name field, type the name of the stereotype tag	
4	In the Type field, click on the drop-down arrow and select the attribute type	
5	In the Initial field, type the initial value of the tag	Add Enumeration Elements ^[1047] (for creating enumerated types for Tagged Values)
6	In the Notes field, type a description of the tag	
7	Click on the Save button and Close button	

Learn More:

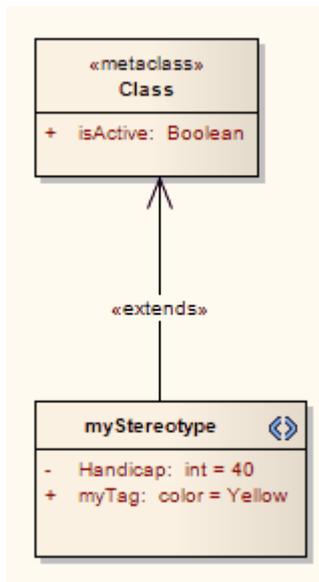
- [Define Stereotype Tags with Predefined Tag Types](#) ^[1045]
- [Define Stereotype Tags with Supported Attributes](#) ^[1045]
- [Use the Tagged Value Connector](#) ^[1046]

Define Predefined Tag Types:

To define a stereotype tag with a predefined Tagged Value Type, you must first create the predefined Tagged Value Type. For full instructions on how to do this, see the *Create Structured Tagged Values* topic.

Assign Predefined Tag Types to Stereotypes:

To assign a predefined tag type to a stereotype, just create an attribute with the same name. For example, to make the Tagged Value *Handicap* appear in a stereotype, create an attribute named *Handicap*. You can set the default value for the Tagged Value by giving the attribute an *Initial* value.

**Learn More:**

- [Create Structured Tagged Values](#)

Supported stereotype attribute tags are special tags that set the default behavior of stereotyped elements, such as the initial size of the element and the default location of any image files associated with the stereotype.

How to:

To define tags for a stereotype with supported attributes, follow the steps below:

Step	Description	See Also
1	Open the Attributes dialog for the stereotyped element	
2	In the Name field, type the name of the stereotype tag	
3	In the Initial field, type the initial value of the tag	
4	For supported attributes you set only the Name (which must match the attributes listed in the supported attributes section) and the Initial value; do not set the other values	

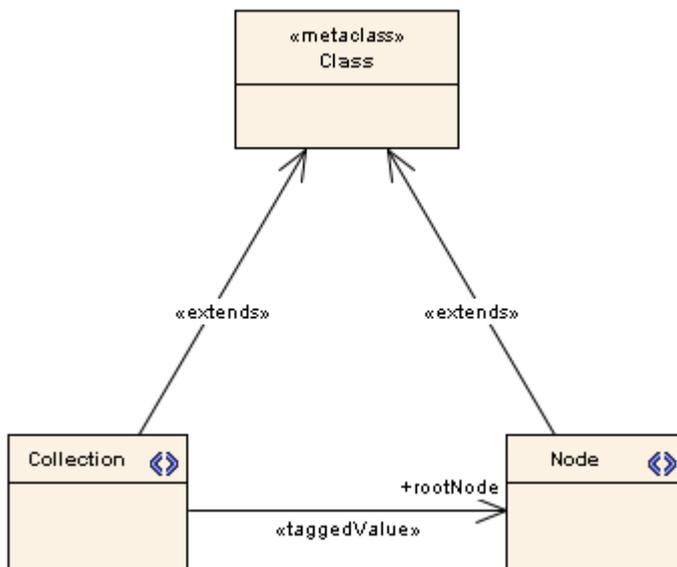
Step	Description	See Also
	Click on the Save button and Close button	

Learn More:

- [Supported Attributes](#) ^[1051]

In a Profile, you can use the *Tagged Value* connector to define a Tagged Value that has as its value the name of an element containing the stereotype pointed to. You select the Tagged Value connector from the Profile pages of the Toolbox.

The following diagram demonstrates how you might use the connector. It shows a (*saved* and *imported*) profile that defines two stereotypes: «*Collection*» and «*Node*». The «*Collection*» stereotype has a Tagged Value connector with the target role named *rootNode*, pointing to the «*Node*» stereotype.



In the Tagged Values window for the connector, against *rootNode*, you click on the selection button ((...)). This displays the Select <Item> dialog, through which you locate the elements in the current model with the «*Node*» stereotype. You can then select one of these elements as the value of the tag.

6.10.1.1.2.4 Define Stereotype Constraints

Defining constraints for stereotypes uses the same procedure as defining constraints for any Class.

How to:

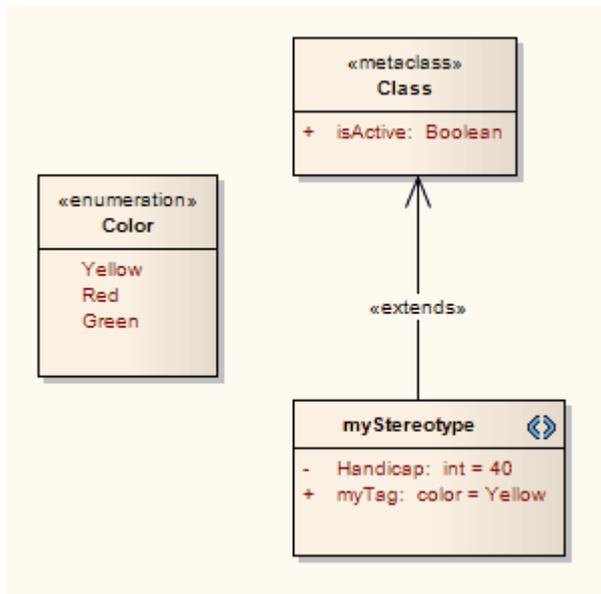
To define constraints for a stereotype, follow the steps below:

Step	Description	See Also
1	Open the Class Properties dialog of the stereotype element in a diagram	
2	Click on the Constraints tab and click on the New button to create a new constraint	
3	In the Constraint field, type the value of the constraint	

Step	Description	See Also
4	In the Type field, click on the drop-down arrow and select the appropriate type	
5	In the Status field, click on the drop-down arrow and select the appropriate status	
6	In the Notes field, type any additional information required	
7	Click on the Save button, and on the OK button to close the dialog	

6.10.1.1.2.5 Add Enumeration Elements

Enumerations can be used to restrict the values available to stereotype tags.



How to:

To add an Enumeration element, follow the steps below:

Step	Description	See Also
1	Open your Profile Package child Class diagram	
2	In the Toolbox, select More tools Profile The contents of the Profile page of the Toolbox display	
3	Drag an <i>Enumeration</i> item from the toolbox onto the diagram If the Properties dialog does not display, double-click on the element on the diagram	
4	In the Name field, type the name of the new Enumeration	
5	Click on the Details tab and on the Attributes button The Attributes Properties dialog displays	
6	In the Name field, type the name of the Enumeration attribute	

Step	Description	See Also
7	In the Type field, click on the drop-down arrow and select the appropriate type	
8	In the Initial field, type the initial value of the attribute	
9	Click on the Save button, and repeat steps 6 to 9 for additional attributes	
10	When you are finished, click on the Close button	
11	Right-click on the <i>Stereotype</i> element and select the Attributes context menu option The Attribute Properties dialog displays for the <i>stereotype</i>	
12	In the Name field type a name for the attribute	
13	In the Type field click on the (...) button and select the name of the enumeration element from the Select <Item> dialog	
14	In the Initial field type the name of the required enumeration attribute as the default value	
15	Click on the Save and Close buttons You have now generated a drop-down list for setting the value of the tag in the Tagged Values window	

6.10.1.1.2.6 Add Shape Scripts

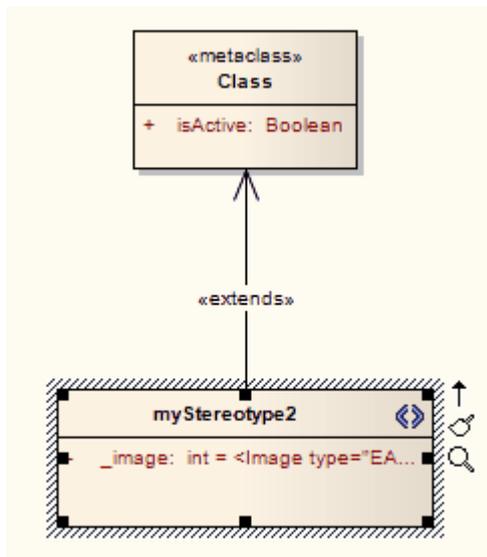
How to:

To add a Shape Script to a stereotype in a UML Profile (such as an MDG Technology Stereotype Profile), follow the steps below:

Step	Description	See Also
1	On the Profile Package child diagram, select a <i>Stereotype</i> element	
2	Right-click on the element and select the Attributes context menu option	
3	In the Attributes Properties dialog, in the Name field, type _image	
4	Click on the (...) button next to the Initial field The Shape Editor dialog displays	Shape Editor 
5	Enter the Shape Script in the Shape Editor dialog Click on the OK and Close buttons	

In some cases, particularly MDG Technology Stereotype Profiles, your Shape Script might include externally-defined images. In this case the Shape Script would include the image method, specifying the image file name prefixed with the technology name.

The Stereotype element now resembles the example below:

**Notes:**

- If you are creating a Shape Script for an Association Class, be aware that the Shape Script is applied to both the Class part and the Association part; therefore, you might have to include logic in the shape main that tests the type of the element so that you can give separate drawing instructions for Class and for Association

Such logic is not necessary in the:

- shape source or shape target, which are ignored by Classes, or the
- decoration shapes which are ignored by Associations

Learn More:

- [MDG Technology Stereotype Profiles](#) ^[1058]
- [Shape Scripts](#) ^[1089]
- [Drawing Methods](#) ^[1094]

6.10.1.1.2.7 Set Default Appearance

You can define the appearance of stereotyped elements and connectors as you create or edit the stereotypes, using the Override Appearance and Default Colors panels of the UML Types dialog. However, an easier way is to review your completed profile diagram and set the default appearance of the elements and connectors in place.

Simply click on the required element or connector and press **(F4)**, then define the background, font and border colors and border thickness as appropriate, on the Default Appearance dialog.

When you save the profile containing the stereotyped elements and connectors, make sure that you select the **Color and Appearance** checkbox on the Save UML Profile dialog.

Learn More:

- [Creating Custom Stereotypes](#) ^[1039]
- [Configure Default Appearance](#) ^[641]
- [Saving Profiles](#) ^[1050]

6.10.1.1.2.8 Export a Profile

Once you have created a Profile and defined the elements and metaclasses, you can save (export) the Profile to disk for future models.

How to:

To save a Profile, follow the steps below:

Step	Description	See Also
1	<p>If your profile is:</p> <ul style="list-style-type: none"> a single profile spread over multiple diagrams within the same Profile package, find the Profile package in the Project Browser window, right-click on it and select the Save Package as UML Profile context menu option one of multiple profiles within the same Profile package, right-click anywhere in the background of the Profile diagram and select the Save as Profile context menu option a single diagram within the Profile package, choose either the Save Package as UML Profile context menu option or the Save as Profile context menu option <p>The Save UML Profile dialog displays</p>	Save Profile Options <small>[1050]</small>
2	<p>Click on the (...) (Browse) button, and select the export destination for the XML Profile file</p> <p>If necessary, edit the profile filename, but do not delete the .xml extension</p>	
3	<p>In the Profile Type field, use the default value EA (UML)2.X (or, if necessary, click on the drop-down arrow and select this value)</p>	
4	<p>Set the required export options for all stereotypes defined in the profile:</p> <ul style="list-style-type: none"> Element Size - select the checkbox to export the element size attributes Color and Appearance - select the checkbox to export the color (background, border and font) and appearance (border thickness) attributes Alternate Image - select the checkbox to export the metafile images Code Templates - select the checkbox to export the code templates, if they exist 	Color and Appearance <small>[1049]</small>
5	<p>Click on the Save button to save the profile to disk</p>	

Learn More:

- For information on importing and using the profile in modeling, see the [UML Profiles](#) [1026] topic.

When you save a UML Profile, you can save it either from the package or from the diagram, depending on whether the Profile is:

- a single profile spread over multiple diagrams within the same Profile package (find the Profile *package* in the Project Browser, right-click on it and select the **Save Package as UML Profile** context menu option), which is typically the case for a stereotypes profile
- one of multiple profiles within the same Profile package (right-click anywhere in the background of the Profile *diagram* and select the **Save as Profile** context menu option); for example, when creating multiple toolbox profiles
- a single diagram within the Profile Package (choose *either* the **Save Package as UML Profile** context

menu option or the **Save as Profile** context menu option).

The two context menu options produce slightly different results. You should take these into consideration, especially in the third instance where you could choose either option.

Save From Diagram	Save From Package	Notes
The profile takes the diagram name	The profile takes the package name	Package and diagram names are not necessarily the same, although you can save a lot of confusion if you make them the same or very similar For example: package <i>GL</i> with diagrams <i>GL1</i> , <i>GL2</i> , <i>GL3</i>
The profile takes the diagram's notes	The profile takes the package's notes	
You can take the default size and appearance (including alternate image) from the diagram object	You cannot take the default size and appearance from the diagram object You can use the <i>_sizeX</i> , <i>_sizeY</i> and <i>_image</i> properties, but there is no equivalent for default colors	
Can be much faster	Can be much slower	The difference arises because diagram objects are kept in memory and Project Browser elements aren't This is only likely to be an issue if the profile is a large one and you are using a slow network connection to a remote repository

6.10.1.1.2.9 Supported Attributes

Supported Stereotype Attributes:

The following attributes can be applied to stereotypes in UML Profiles:

Attribute	Meaning	See Also
icon	Contains the path to a bitmap file to be used as the Project Browser icon for all elements other than <i>Package</i> , with the given stereotype The bitmap must be 16x16 pixels For a transparent background, use light grey - RGB(192,192,192) For this attribute to work correctly, the <i>_metatype</i> attribute must also be used (see below)	
_image	Shape script definition	
_instanceMode	Used for defining behavior on creating an instance	Defining Behavior on Creating an Instance <small>[1055]</small>
_instanceOwner		

Attribute	Meaning	See Also
_instanceType		
_metatype	Used for defining stereotypes as metatypes	Defining Stereotypes as Metatypes ^[1054]
_sizeY	Initial height of the element, in pixels at 100% zoom	
_sizeX	Initial width of the element, in pixels at 100% zoom	
_strictness	Used for restricting application of multiple stereotypes	Restricting Application of Multiple Stereotypes ^[1054]

Supported Metaclass Attributes:

The following attributes can be applied to metaclass Classes in UML Profiles, and refer to the stereotypes that extend them:

Attribute	Meaning	See Also
_AttInh	If set to 1 , switches on the <i>Inherited Features: Show Attributes</i> setting	
_AttPkg	If set to 1 , switches on the <i>Attribute Visibility: Package</i> setting	
_AttPri	If set to 1 , switches on the <i>Attribute Visibility: Private</i> setting	
_AttPro	If set to 1 , switches on the <i>Attribute Visibility: Protected</i> setting	
_AttPub	If set to 1 , switches on the <i>Attribute Visibility: Public</i> setting	
_ConInh	If set to 1 , switches on the <i>Show Element Compartments: Inherited Constraints</i> setting	
_Constraint	If set to 1 , switches on the <i>Show Element Compartments: Constraints</i> setting	
_DefaultDiagramType	Used for defining child diagram types	Defining Child Diagram Types ^[1057]
_HideStyle	If set to a comma-separated list of stereotypes, sets the <i>Hide Stereotyped Features</i> filter	
_isVertical	If set to True for an ActivityPartition stereotype, creates a vertical ActivityPartition by default	
_lineStyle	Sets the line style of a connector; the value of the attribute can be one of: <ul style="list-style-type: none"> • direct • auto • custom • bezier • treeH (horizontal) • treeV (vertical) • treeLH (lateral horizontal) 	

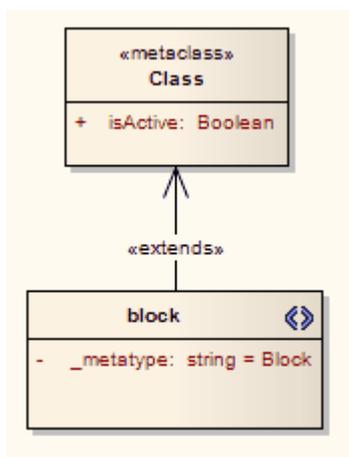
Attribute	Meaning	See Also
	<ul style="list-style-type: none"> • treeLV (lateral vertical) • orthogonalS (orthogonal, square corners) • orthogonalR (orthogonal, rounded corners) 	
_MakeComposite	Used for creating composite elements	Creating Composite Elements ^[1056]
_OpInh	If set to 1 , switches on the <i>Inherited Features: Show Operations</i> setting	
_OpPkg	If set to 1 , switches on the <i>Operation Visibility: Package</i> setting	
_OpPri	If set to 1 , switches on the <i>Operation Visibility: Private</i> setting	
_OpPro	If set to 1 , switches on the <i>Operation Visibility: Protected</i> setting	
_OpPub	If set to 1 , switches on the <i>Operation Visibility: Public</i> setting	
_PType	If set to 1 , switches on the <i>Show element type (Port or Part only)</i> setting	
_ResInh	If set to 1 , switches on the <i>Show Element Compartments: Inherited Responsibilities</i> setting	
_Responsibility	If set to 1 , switches on the <i>Show Element Compartments: Responsibilities</i> setting	
_Runstate	If set to 1 , switches on the <i>Hide Object Runstate in current diagram</i> setting	
_SourceAggregation	Used to set the aggregation type at the end of a connector; do not set <i>both</i> _SourceAggregation and _TargetAggregation Set to 1 for shared, 2 for composite	
_SourceMultiplicity	Used to set the multiplicity of the source element, such as 1..* or 0..1	
_SourceNavigability	If the connector is non-navigable, set this attribute to Non-Navigable ; for other values, set the direction attribute	Direction Attribute ^[191]
_SubtypeProperty	Used to specify the fully qualified name of a Tagged Value that acts as a subtype for the element, and that must be set when created This Tagged Value must be created on the stereotype using an Association to an Enumeration element	
_Tag	If set to 1 , switches on the <i>Show Element Compartments: Tags</i> setting	
_tagGroupings	Maps the Tagged Values into the _tagGroups (below) displayed in the Tagged Values window in the form: <i>tagName1=groupName1;tagName2=groupName2;</i>	
_tagGroups	A comma-separated list of required groups in the order in which they are to be displayed in the Tagged Values window	
_tagGroupStates	Maps _tagGroups displayed in the Tagged Values window to the state of open or closed , in the form: <i>group1=open;group2=closed;</i>	
_TagInh	If set to 1 , switches on the <i>Show Element Compartments: Inherited Tags</i> setting	
_TargetAggregation	Used to set the aggregation type at the end of a connector; do not	

Attribute	Meaning	See Also
tion	set <i>both</i> _SourceAggregation and _TargetAggregation Set to 1 for shared, 2 for composite	
_TargetMultiplicity	Used to set the multiplicity of the target element, such as 1..* or 0..1	
_TargetNavigability	If the connector is non-navigable, set this attribute to Non-Navigable ; for other values, set the Direction attribute	Direction Attribute ^[1917]

The *_metatype* attribute is applied to a stereotype element. This is used where users want to hide the identity of an element as a stereotyped UML element. It is also a method of getting custom types to appear in contexts where only Enterprise Architect's inbuilt types would normally appear; for example in the lists of element types in the Relationship Matrix.

The **Strict UML Syntax check** is not performed for stereotyped elements that have had the *_metatype* attribute set.

In the following example from SysML, *block* is defined as a stereotype that extends a UML Class.



However, a SysML user isn't interested in UML Classes, only in SysML Blocks. An element created from a stereotype defined this way, while behaving like a stereotyped Class in most contexts:

- Shows *Block Properties* rather than *Class Properties* as the title of its Properties dialog
- Is auto-numbered as *Block 1* not *Class 1* on creation, and
- Appears as *Block* not *Class* in many other contexts throughout Enterprise Architect

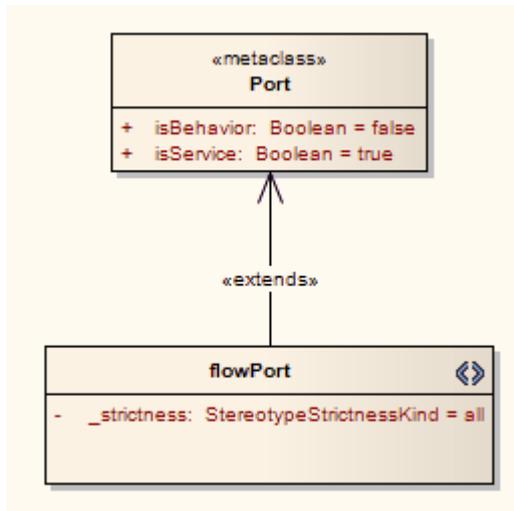
Learn more:

- [Diagram](#) ^[426] (Strict UML Syntax Check)

The **_strictness** attribute is applied to a stereotype element. It defines to what level multiple stereotypes can be applied to an element. The type of the attribute is *StereotypeStrictnessKind* and it can have one of four values:

- *profile*, which states that an element of this type cannot be given more than one different stereotype from the same profile
- *technology*, which states that an element of this type cannot be given more than one different stereotype from the same technology
- *all*, which states that an element of this type cannot have multiple stereotypes at all, or
- *none*, which is the default Enterprise Architect behaviour and states that there are no restrictions on the use of multiple stereotypes

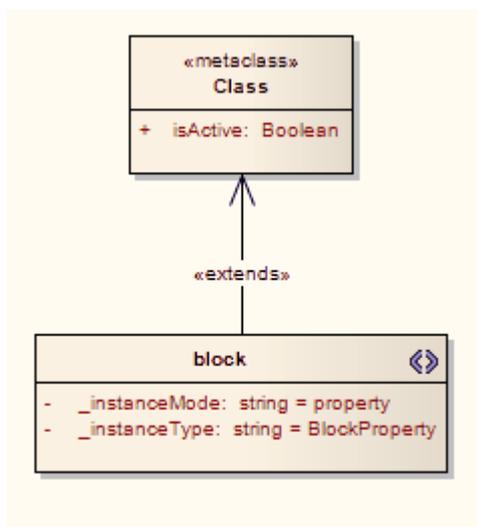
The following example is from SysML and shows that a «flowPort» cannot have any other stereotype applied to it.



Reference:

Attribute	Meaning	See Also
_instanceType	Applied to a stereotype element, defining what kind of element is created as an instance of this element type The value corresponds to the metatype given to a stereotype using the _metatype attribute It is shown on the Paste Element dialog and is translated if it matches an Enterprise Architect element type	
_instanceMode	Applied to a stereotype element, controlling the text in the Paste Element dialog after being translated Valid values are instance and property , with the default being instance	
_instanceOwner	Applied to a stereotype element, controlling the text in the Paste Element dialog It is translated if it matches an Enterprise Architect element type; the default value is Element	

The following example from SysML shows that when an instance of a Block is created, it is created as a **BlockProperty** element.



Learn More:

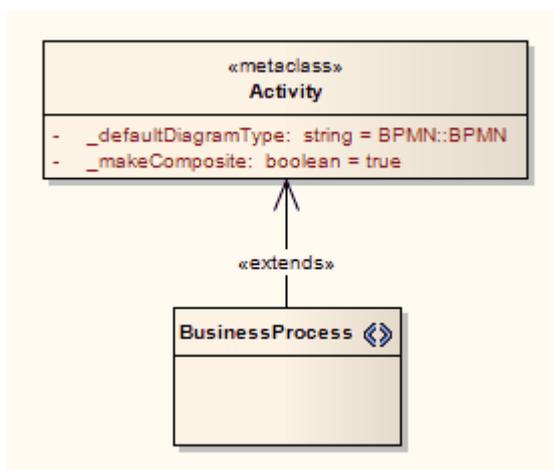
- [Paste Element from the Project Browser](#)^[576]

The **_makeComposite** attribute is applied to a metaclass element, not a stereotype element. It defines whether an element is always made composite when created.

Unless you also use the **_defaultDiagramType** attribute to define the child diagram type, the child diagram created is the usual default diagram type for the metaclass.

A stereotyped package is not by default created with a child diagram, so you should use the **_makeComposite** attribute to ensure the child diagram is created.

The following example from BPMN shows that a *BusinessProcess* element is always created as a Composite element with a BPMN custom child diagram.



Learn More:

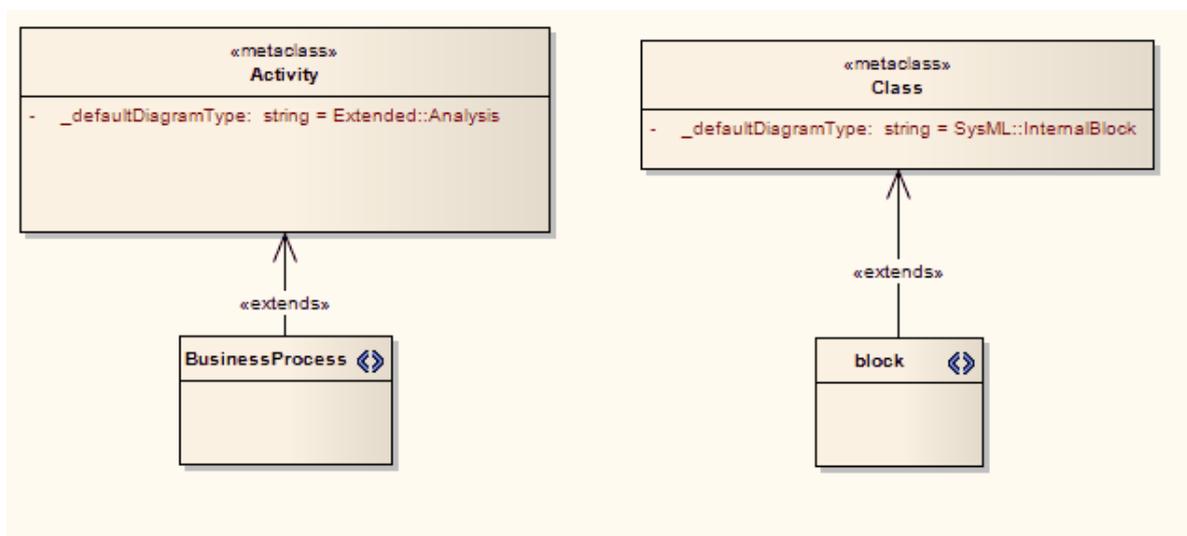
- [Defining Child Diagram Types](#)^[1057]

The `_defaultDiagramType` attribute is applied to a metaclass element, not a stereotype element. It defines the type of diagram created when an element is made composite.

This attribute can take as its name any of the inbuilt diagram types of Enterprise Architect as listed below.

Alternatively, if a custom diagram type is required, it should be prefixed with the *diagram profile name* and ':'. The *diagram profile name* is the name given to the profile when you save it, which by default is the name of the profile package or profile diagram. If you follow the recommendation in Create Diagram Profiles, the diagram profile name is based on the technology name, but be aware that the attribute prefix is **not a direct** reference to the technology name.

The following examples show a `«BusinessProcess»Activity` that, when made a composite element, automatically creates an Analysis diagram, and a `«block»` stereotype that creates a SysML *InternalBlock* custom diagram.



You can also use the `_defaultDiagramType` attribute for packages, extending the Package metaclass.

Initial Values:

The following initial values for `_defaultDiagramType` should be used to refer to Enterprise Architect's inbuilt diagram types:

- UML Behavioral::Use Case
- UML Behavioral::Activity
- UML Behavioral::State Machine
- UML Behavioral::Communication
- UML Behavioral::Sequence
- UML Behavioral::Timing
- UML Behavioral::Interaction Overview
- UML Structural::Package
- UML Structural::Class
- UML Structural::Object
- UML Structural::Composite Structure
- UML Structural::Component
- UML Structural::Deployment
- Extended::Custom
- Extended::Requirements
- Extended::Maintenance

- Extended::Analysis
- Extended::User Interface
- Extended::Data Modeling
- Extended::ModelDocument.

Learn More:

- [Creating Composite Elements](#) ^[1056]
- [Custom Diagram Types](#) ^[1080]

6.10.1.1.2.10 Stereotype Profiles

Customized stereotypes should be contained in one or more profiles. The stereotypes within each profile use the profile name as the namespace. You then add the profiles into an MDG Technology.

Create one or more packages with the «*profile*» stereotype, each package name being the namespace. Within each package create profile diagrams defining all the stereotypes in the namespace. You can use multiple diagrams to do this, but do not use nested packages.

Give the «*profile*» package a description in the **Notes** field, such as *MDG Technology for BPMN*. When all of the stereotypes are defined (make sure that every stereotype extends at least one *Metaclass*) right-click on the profile package in the Project Browser and select the **Save Package as UML Profile** context menu option, then proceed as usual.

To define an MDG technology's Toolbox pages, you create a separate profile.

Learn More:

- [Save Package as a UML Profile](#) ^[1050]
- [Add Profile to an MDG Technology](#) ^[1068]
- [Toolbox Profiles](#) ^[1074]

6.10.1.1.3 Quick Linker

Topics:

Topic	Discussion	See Also
Introduction	<p>The Quick Linker provides a fast and simple way to create new elements and connectors on a diagram</p> <p>When an element is selected in a diagram, the Quick Linker arrow is displayed in the upper right corner of a element; simply clicking and dragging the arrow enables you to create new connectors and elements</p> <p>The philosophy behind the built-in Quick Linker definitions is to provide, not a complete list of valid or legal connections, but a short and convenient list of the commonest connections for the given context</p> <p>As part of a UML Profile, you can add to or replace the built-in Quick Linker definitions</p>	<p>Quick Linker Definition Format ^[1059]</p> <p>Quick Linker Example ^[1061]</p> <p>Hide Default Quick Linker Settings ^[1062]</p> <p>Quick Linker Object Names ^[1062]</p>
Customized Quick Linker Settings	<p>A Quick Linker definition is a Comma Separated Value (CSV) format file; it is best manipulated in a spreadsheet, which should be set up to save the CSV file as comma-separated text without quotation marks around text fields</p>	

Topic	Discussion	See Also
	<p>To add a Quick Linker definition file to a profile or technology, simply place a <i>DocumentArtifact</i> element onto the Profile diagram, give it the name <i>QuickLink</i>, then double-click on it</p> <p>Open your CSV file in a text editor such as Notepad and copy and paste the contents into the <i>DocumentArtifact</i> element; the definitions are saved with the profile and are processed and applied when the profile is imported</p> <p>The same applies if a profile is included within a technology, with the proviso that the QuickLink element must be in the same profile as the link stereotype definitions; this means that a technology could have a set of Quick Link definitions for each profile</p>	

6.10.1.1.3.1 Quick Linker Definition Format

A Quick Linker definition is a text file consisting of records terminated by new-line characters. Each record must consist of 23 comma-separated fields, as defined by the table below. The values of each field must not be in quotes (" ").

A Quick Linker definition can include comments: all lines that begin with // are ignored by Enterprise Architect.

Each record of the Quick Linker definition represents a single entry on the Quick Linker menu. Some fields define the menu command; some fields can be thought of as filters, with the entry being ignored if the filter condition isn't met.

A Quick Linker definition has the following fields.

Column	Field	Description
A	Source Element Type	The row is ignored unless a connector is being dragged away from this type of element
B	Source Stereotype Filter	If set, the row is ignored unless a connector is being dragged away from an element with this stereotype
C	Target Element Type	If set, the row is ignored unless a connector is being dragged onto this type of element If blank, the row is ignored unless a connector is being dragged onto an empty piece of diagram
D	Target Stereotype Filter	If set and Target Element Type is also set, the row is ignored unless a connector is being dragged onto an element with this stereotype
E	Diagram Filter	Contains either an inclusive or exclusive list of built-in diagrams ^[108] , which limits the diagrams the given kind of connector can be included on Each diagram name is terminated by a semi-colon; excluded diagram names are preceded by an exclamation mark Example of an inclusive list: <i>Collaboration;Object;Custom;</i> Example of an Exclusive list: <i>!Sequence;</i>
F	New Element Type	If set and Create Element is also set, results in the creation of an element of this type
G	New Element	If set and Create Element is also set, results in the creation of an

Column	Field	Description
	Stereotype	element with this stereotype
H	New Link Type	If set and Create Link is also set, results in the creation of a connector of this type
I	New Link Stereotype	If set and Create Link is also set, results in the creation of a connector with this stereotype
J	New Link Direction	Can be: <ul style="list-style-type: none"> • directed (always creates an association from source to target) • from (always creates an association from target to source) • undirected (always creates an association with unspecified direction) • bidirectional (always creates a bi-directional association), or • to (creates either a directed or undirected association, depending on the value of the Association Direction option) <p>Not all of these work with all connector types; for example, you cannot create a bi-directional Generalization</p>
K	New Link Caption	If a new connector is being created but not a new element, then this is the text that appears on the context menu
L	New Link & Element Caption	If a new connector AND a new element are being created, then this is the text that appears on the context menu
M	Create Link	If set to TRUE , results in creation of a new connector; otherwise should be left blank
N	Create Element	If set to TRUE the row is ignored unless a connector is being dragged onto an empty piece of diagram and results in creation of a new element; otherwise should be left blank This overrides the values of Target Element Type and Target Stereotype Filter
O	Disallow Self connector	Should be set to TRUE if self connectors are invalid for this kind of connector; otherwise should be left blank
P	Exclusive to ST Filter + No inherit from Metatype	If set to TRUE , indicates that elements of type <i>Source Element Type</i> with the stereotype <i>Source Stereotype Filter</i> do not display the Quick Linker definitions of the equivalent unsteretyped element
Q	Menu Group	If set, indicates the name of a sub-menu in which a menu item is created
R	Complexity Level	Not implemented, always set to 0
S	Target Must Be Parent	If set to TRUE this menu item only appears when dragging from a child element to its parent; for example from a port to its containing Class
T	Embed element	If set to TRUE the element being created is embedded in the target element; otherwise should be left blank
U	Precedes Separator LEAF	If set to TRUE results in a menu separator being added to the Quick Linker menu; otherwise should be left blank
V	Precedes Separator GROUP	If set to TRUE results in a menu separator being added to the Quick Linker sub-menu; otherwise should be left blank
W	Dummy Column	Depending on which spreadsheet application you use, this column

Column	Field	Description
		might require a value in every cell to force CSV export to work correctly with trailing blank values

6.10.1.1.3.2 Quick Linker Example

This example uses a Class element with the stereotype «quick». The example scenario is this: when you drag a connector away from one of these elements, you want to create a Dependency either to or from a component element. When you drag a connector onto an existing Port or component element, you want a Dependency either to or from the component or, in the case of a component, you want to be able to create an embedded Port element.

This results in 8 records in the [Quick Linker definition](#) file.

1. Dependency to new Component
2. Dependency from new Component
3. Dependency to existing Component
4. Dependency from existing Component
5. Dependency to existing Port
6. Dependency from existing Port
7. Dependency to existing Component, create new Port
8. Dependency from existing Component, create new Port

In the spreadsheet, this is implemented by the following values:

	A	B	C	E	F	H	J	K
1	//Source Element Type	Source ST filter	Target Element Type	Diagram Filter	New Element Type	New Link Type	New Link Direction	New Link Caption
2	Class	quick			Component	Dependency	to	
3	Class	quick			Component	Dependency	from	
4	Class	quick	Component			Dependency	to	Dependency to
5	Class	quick	Component			Dependency	from	Dependency from
6	Class	quick	Port			Dependency	to	Dependency to
7	Class	quick	Port			Dependency	from	Dependency from
8	Class	quick	Component		Port	Dependency	to	
9	Class	quick	Component		Port	Dependency	from	

	L	M	N	O	P	Q	R	S	T	U	V
1	New Link & Element Caption	Create Link	Create Element	Disallow Self connector	No inherit from metatype	Menu Group	Complexity Level	Target Must Be Parent	Embed element	Preceeds Separator LEAF	Preceeds Separator GROUP
2	Dependency to	TRUE	TRUE	TRUE	TRUE	Component	0				
3	Dependency from	TRUE	TRUE	TRUE	TRUE	Component	0			TRUE	
4		TRUE		TRUE	TRUE		0				
5		TRUE		TRUE	TRUE		0			TRUE	
6		TRUE		TRUE	TRUE		0				
7		TRUE		TRUE	TRUE		0			TRUE	
8	Dependency to	TRUE	TRUE	TRUE	TRUE	Port	0		TRUE		
9	Dependency from	TRUE	TRUE	TRUE	TRUE	Port	0		TRUE	TRUE	

This saves to the following CSV:

```

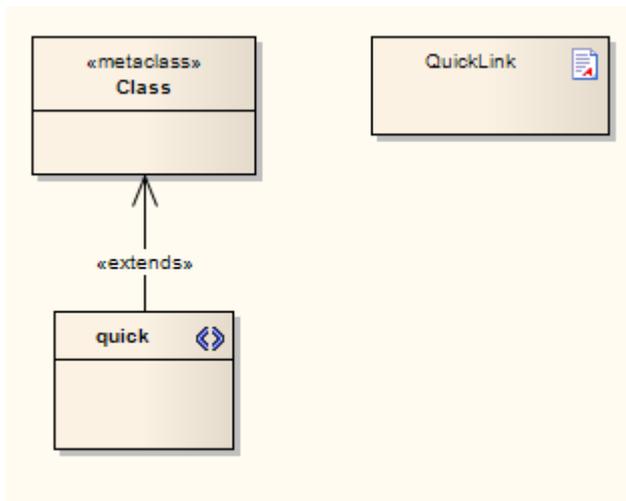
Class, quick, , , Component, , Dependency, , to, , Dependency to, TRUE, TRUE, TRUE,
TRUE, Component, 0, , , ,
Class, quick, , , Component, , Dependency, , from, , Dependency from, TRUE, TRUE,
TRUE, TRUE, Component, 0, , TRUE, ,
Class, quick, Component, , , , Dependency, , to, Dependency to, , TRUE, , TRUE,
    
```

```

TRUE, , 0, , , , ,
Cl ass, qui ck, Component, , , , , Dependency, , from, Dependency from, , TRUE, , TRUE,
TRUE, , 0, , , TRUE, ,
Cl ass, qui ck, Port, , , , , Dependency, , to, Dependency to, , TRUE, , TRUE,
TRUE, , 0, , , , ,
Cl ass, qui ck, Port, , , , , Dependency, , from, Dependency from, , TRUE, , TRUE,
TRUE, , 0, , , TRUE, ,
Cl ass, qui ck, Component, , , Port, , Dependency, , to, , Dependency to, TRUE, TRUE,
TRUE, TRUE, Port, 0, , TRUE, , ,
Cl ass, qui ck, Component, , , Port, , Dependency, , from, , Dependency from, TRUE,
TRUE, TRUE, TRUE, Port, 0, , TRUE, TRUE, ,

```

You can create the following profile and cut and paste the CSV data into the document artifact to test the effect.



6.10.1.1.3.3 Hide Default Quick Linker Settings

If you have a Quick Linker definition with the *Exclusive to stereotype* flag (column P) set to **TRUE**, then the default Quick Linker definitions between the given source and target are overridden. However, you might want to override the defaults without actually having a Quick Linker definition. For example, if you don't define any Quick Links for a «quick» Class to another «quick» Class, Enterprise Architect displays the default Quick Links for a Class to another Class. To override this behaviour, create a Quick Linker definition that has the source element type, source stereotype filter, target element type and target stereotype filter fields (columns A, B, C and D) all set, with the **Exclusive to stereotype** flag (column P) set to **TRUE**, and with the new link type field (column H) set to **<none>**.

For example, add this line to the example in [Quick Linker Example](#)^[1061]:

```
Cl ass, qui ck, I n t e r f a c e, , , , , <none>, , , , , , TRUE, , 0, , , , ,
```

This overrides the default Class-to-Interface Quick Links when a Quick Link is dragged from a «quick» Class to an Interface element.

This technique does not affect the automatic appearance of Dependency, Trace, Information Flow and Help items on the Quick Linker menu.

6.10.1.1.3.4 Quick Linker Object Names

List of Element Types:

The following element names can be used in Quick Linker definitions:

Action	ExecutionEnvironment	Package
ActionPin	ExitPoint	Part
Activity	ExitState	Port
ActivityParameter	ExpansionNode	PrimitiveType
ActivityPartition	ExpansionRegion	ProvidedInterface
Actor	Feature	Receive
Artifact	GUIElement	RequiredInterface
Boundary	HistoryState	Requirement
CentralBufferNode	InformationItem	Screen
Change	InitialActivity	Send
ChoiceState	InitialState	Sequence
Class	InteractionOccurrence	Signal
Collaboration	Interface	State
Component	Issue	StateLifeline
Data Type	InterruptableActivityRegion	StateMachine
Decision	JunctionState	Synchronization_H
DeepHistoryState	MergeNode	Synchronization_V
Deployment Specification	MessageEndpoint	SynchState
Device	n-ary Association	UMLDiagram
DiagramGate	Node	UseCase
EntryPoint	Object	ValueLifeline
EntryState	ObjectNode	

List of Connector Types:

The following connector names can be used in Quick Linker definitions:

Aggregation	Deployment	PackageMerge
Association	Extension	Realization
AssociationClass	Generalization	Redefinition
CommunicationPath	InformationFlow	Sequence
Composition	InterfaceLink	StateFlow
ConnectorLink	Manifest	TemplateBinding
ControlFlow	Nesting	UCExtends
DelegateLink	ObjectFlow	UCIncludes
Dependency	PackageImport	UseCase

6.10.1.2 MDG Technologies - Creating

The Model Driven Generation (MDG) Technologies enable Enterprise Architect users to access and use resources pertaining to a specific technology in Enterprise Architect. There are various options for an administrator or individual user to bring MDG Technologies into use with Enterprise Architect. You should read the first two MDG Technology topics (below) to understand how MDG Technologies are accessed and used within Enterprise Architect.

A further option is that Technology Developers can develop new MDG Technologies and deploy them to the project team as appropriate, as described in the remaining topics listed below.

Learn More:

- [MDG Technologies](#) ^[1031]
- [Manage MDG Technologies](#) ^[1033]
- [Create MDG Technologies](#) ^[1064]
- [Working With MTS Files](#) ^[1073]
- [Customize Toolbox Profiles](#) ^[1074]
- [Create Diagram Profiles](#) ^[1080]
- [Create Learning Center Profiles](#) ^[1082]
- [Define Validation Configuration](#) ^[1086]
- [Incorporate Model Templates](#) ^[1087]
- [Deploy an MDG Technology](#) ^[1089]

An example of creating an MDG Technology for an Enterprise Architecture framework is provided in: [Enterprise Architecture Framework Design with Sparx Systems Enterprise Architect](#).

[Set Up Technology Element Images](#) ^[1088] describes the process of ensuring that a technology element is represented by a specific image on a diagram, when the technology is deployed in a model.

6.10.1.2.1 Create MDG Technologies

Using the MDG Technology Wizard, you can create MDG Technology files that can include UML Profiles, code modules, Patterns, images, Tagged Value Types, RTF report templates, linked document templates, Toolbox pages and Learning Center pages.

This facility is not available in the Enterprise Architect Desktop edition.

Access: **Tools | Generate MDG Technology File**

How To:

To create an MDG Technology file, follow the steps below:

Step	Description	See Also
1	Select the Generate MDG Technology File menu option The MDG Technology Creation Wizard screen displays	
2	Click on the Next button The MDG Technology Wizard prompts you to either: <ul style="list-style-type: none"> • Create an MDG Technology File by creating a new MDG Technology Selection (MTS) file 	

Step	Description	See Also																				
	<ul style="list-style-type: none"> • Create an MDG Technology File using an existing MTS file, or • Not use any MTS file <p>(An MTS file stores the selected options that you define during the creation of an MDG Technology File; if you use an MTS file, you can later modify it to add or remove specific items in the MDG Technology File, which is the recommended process)</p>																					
3	<p>Select the appropriate MTS file option</p> <p>Click on the Next button</p> <p>If you selected an MTS file, the MDG Technology Wizard prompts you to save the changes in the existing MTS file or into a new MTS file; this enables you to create a modification based on the existing MTS file, while preserving the original file</p>																					
4	<p>If necessary, type in or browse for the required file path and name</p> <p>Click on the Next button</p> <p>The MDG Technology Wizard - Create screen displays</p>																					
5	<p>Complete the fields on this screen as follows:</p> <table border="1" data-bbox="384 947 1182 1910"> <thead> <tr> <th data-bbox="384 947 555 1003">Option</th> <th data-bbox="555 947 1182 1003">Action</th> </tr> </thead> <tbody> <tr> <td data-bbox="384 1003 555 1059">Technology</td> <td data-bbox="555 1003 1182 1059">Type the name of the MDG Technology</td> </tr> <tr> <td data-bbox="384 1059 555 1171">Filename</td> <td data-bbox="555 1059 1182 1171">Type or select the path and filename of the MDG Technology File The file extension for this file is .xml</td> </tr> <tr> <td data-bbox="384 1171 555 1249">ID</td> <td data-bbox="555 1171 1182 1249">Type a reference for the MDG Technology File, up to 12 characters long</td> </tr> <tr> <td data-bbox="384 1249 555 1305">Version</td> <td data-bbox="555 1249 1182 1305">Type the version number of the MDG Technology File</td> </tr> <tr> <td data-bbox="384 1305 555 1462">Icon</td> <td data-bbox="555 1305 1182 1462">(Optional) Type or select the path and file name of the graphic containing the technology icon The icon is a 16x16 bitmap image that is shown in the list of technologies on the left of the MDG Technologies dialog</td> </tr> <tr> <td data-bbox="384 1462 555 1630">Logo</td> <td data-bbox="555 1462 1182 1630">(Optional) Type or select the path and file name of the graphic containing the technology logo The logo is a 64x64 bitmap image that is shown in the dialog on the top-right corner of the MDG Technologies dialog</td> </tr> <tr> <td data-bbox="384 1630 555 1742">URL</td> <td data-bbox="555 1630 1182 1742">(Optional) If you have any website product information that is helpful for users of this Technology, type or paste the URL in this field</td> </tr> <tr> <td data-bbox="384 1742 555 1865">Support</td> <td data-bbox="555 1742 1182 1865">(Optional) If you have any web-based or other support contact information that might be helpful for users of this Technology, type or paste the contact address in this field</td> </tr> <tr> <td data-bbox="384 1865 555 1910">Notes</td> <td data-bbox="555 1865 1182 1910">Type a short explanation of the functionality of the MDG Technology</td> </tr> </tbody> </table>	Option	Action	Technology	Type the name of the MDG Technology	Filename	Type or select the path and filename of the MDG Technology File The file extension for this file is .xml	ID	Type a reference for the MDG Technology File, up to 12 characters long	Version	Type the version number of the MDG Technology File	Icon	(Optional) Type or select the path and file name of the graphic containing the technology icon The icon is a 16x16 bitmap image that is shown in the list of technologies on the left of the MDG Technologies dialog	Logo	(Optional) Type or select the path and file name of the graphic containing the technology logo The logo is a 64x64 bitmap image that is shown in the dialog on the top-right corner of the MDG Technologies dialog	URL	(Optional) If you have any website product information that is helpful for users of this Technology, type or paste the URL in this field	Support	(Optional) If you have any web-based or other support contact information that might be helpful for users of this Technology, type or paste the contact address in this field	Notes	Type a short explanation of the functionality of the MDG Technology	
Option	Action																					
Technology	Type the name of the MDG Technology																					
Filename	Type or select the path and filename of the MDG Technology File The file extension for this file is .xml																					
ID	Type a reference for the MDG Technology File, up to 12 characters long																					
Version	Type the version number of the MDG Technology File																					
Icon	(Optional) Type or select the path and file name of the graphic containing the technology icon The icon is a 16x16 bitmap image that is shown in the list of technologies on the left of the MDG Technologies dialog																					
Logo	(Optional) Type or select the path and file name of the graphic containing the technology logo The logo is a 64x64 bitmap image that is shown in the dialog on the top-right corner of the MDG Technologies dialog																					
URL	(Optional) If you have any website product information that is helpful for users of this Technology, type or paste the URL in this field																					
Support	(Optional) If you have any web-based or other support contact information that might be helpful for users of this Technology, type or paste the contact address in this field																					
Notes	Type a short explanation of the functionality of the MDG Technology																					
6	<p>Click on the Next button</p> <p>The MDG Technology Wizard - Contents screen displays</p>																					

Step	Description	See Also
7	<p>Select the checkbox for each item to be included in the MDG Technology file</p> <p>Each selection runs specific dialogs to enable definition of the specific items to be included in the MDG Technology, as described in the following topics:</p> <ul style="list-style-type: none"> • Add a Profile ^[1066] • Add a Pattern ^[1067] • Add a Diagram Profile ^[1068] • Add a Toolbox Profile ^[1068] • Add Learning Center Pages ^[1069] • Add Tagged Value Types ^[1069] • Add Code Modules ^[1070] • Add MDA Transformations ^[1071] • RTF Report Templates ^[1072] • Linked Document Templates ^[1072] • Add Images ^[1071] • Add Scripts ^[1072] (Corporate and 'Suite' editions). 	
8	<p>Work through the dialogs displayed in response to your choices, and when all are complete, click on the Next button</p> <p>The MDG Technology Wizard - Finish screen displays, providing information on the items included in the MDG Technology File</p>	
9	<p>If you have used an MTS file and want to update it, select the Save to MTS checkbox</p>	
10	<p>If you are satisfied with the selection of items, click on the Finish button</p> <p>You can now edit the MTS file, if required, to add further items such as:</p> <ul style="list-style-type: none"> • Model Search definitions • Model Views • Model Validation configurations • Model Templates <p>To make the MDG Technology File accessible to an Enterprise Architect model, you must <i>add</i> the technology file path to the MDG Technologies - Advanced dialog</p>	<p>Working with MTS Files ^[1073]</p> <p>Access Remote MDG Technologies ^[1034]</p>

6.10.1.2.1 Add a Profile

When creating an MDG Technology file, you can include UML 2.3-compliant profiles that you have defined.

To use the Profiles section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	<p>Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Profiles checkbox</p>	<p>Creating MDG Technologies, Step 6 ^[1065]</p>

Step	Description	See Also
	The MDG Technology Wizard - Profile files selection dialog displays	
2	In the Directory field, navigate to the directory containing the required Profile or Profiles The Profile files are automatically listed in the Available Files panel	
3	To select each required Profile individually, highlight the Profile in the Available Files list and click on the --> button The file name displays in the Selected Files list Alternatively: To select all available Profiles, click on the -->> button <ul style="list-style-type: none"> • DO NOT select diagram profiles or toolbox profiles on this dialog; this would generate conflicting commands in the .MTS file • Make sure you do include your stereotype profile 	Stereotype Profile ^[1058]
4	Click on the Next button to proceed	

Learn More:

- [Working With Profiles](#) ^[1041]

6.10.1.2.1.2 Add a Pattern

When creating an MDG Technology file, you can include patterns.

To use the Patterns section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Patterns checkbox. The MDG Technology Wizard - Pattern files selection dialog displays.	Creating MDG Technologies, Step 6 ^[1065]
2	In the Directory field, navigate to the directory containing the required pattern or patterns. The pattern files are automatically listed in the Available Files panel.	
3	To select each required pattern individually, highlight the pattern in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available patterns, click on the -->> button.	
4	Click on the Next button to proceed.	

Learn More:

- [Creating Patterns](#) ^[1022]

6.10.1.2.1.3 Add a Diagram Profile

When creating an MDG Technology file, you can include a diagram profile that you have defined.

To use the diagram profiles section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Diagram Types checkbox. The MDG Technology Wizard - Diagram Types dialog displays.	Creating MDG Technologies, Step 6 <small>[1065]</small>
2	In the Directory field, navigate to the directory containing the required diagram profiles. The profiles in the directory are automatically listed in the Available Files panel.	
3	To select each required diagram profile individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available profiles (assuming they are all diagram profiles), click on the -->> button.	
4	Click on the Next button to proceed.	

Learn More:

- [Custom Diagram Types](#)
[1080]

6.10.1.2.1.4 Add a Toolbox Profile

When creating an MDG Technology file, you can include Enterprise Architect **Toolbox** page definitions that you have created.

To use the Toolboxes section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Toolboxes checkbox. The MDG Technology Wizard - Toolboxes dialog displays.	Creating MDG Technologies, Step 6 <small>[1065]</small>
2	In the Directory field, navigate to the directory containing the required toolbox profiles. The profile files are automatically listed in the Available Files panel.	
3	To select each required toolbox profile individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available profiles (assuming they are all toolbox profiles), click on the -->> button.	
4	Click on the Next button to proceed.	

Learn More:

- [Custom Toolboxes](#)
[1074]

6.10.1.2.1.5 Add Learning Center Pages

When creating an MDG Technology file, you can include Enterprise Architect Learning Center profiles that you have created.

To use the Taskpages section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Taskpages checkbox. The MDG Technology Wizard - Learning Center dialog displays.	Creating MDG Technologies, Step 6 <small>[1065]</small>
2	In the Directory field, navigate to the directory containing the required taskpage profiles. The profile files are automatically listed in the Available Files panel	
3	To select each required learning center profile individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available profiles (assuming they are all Learning Center profiles), click on the -->> button.	
4	Click on the Next button to proceed.	

Learn More:

- [Custom Learning Center](#)
[1082]

6.10.1.2.1.6 Add Tagged Value Types

When creating an MDG Technology file, you can include Tagged Value Types.

To use the Tagged Value Types section of the MDG Technology Types Wizard, follow the steps below:

Step	Description	See Also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Tagged Value Types checkbox. The MDG Technology Wizard - Tagged Value Types dialog displays.	Creating MDG Technologies, Step 6 <small>[1065]</small>
2	To select each required Tagged Value Type individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available Tagged Value Types, click on the -->> button.	
3	Click on the Next button to proceed.	

Learn More:

- [Tagged Value Types](#)
[1108]

6.10.1.2.1.7 Add Code Modules

When creating an MDG Technology file, you can include code modules.

To use the code modules section of the MDG Technology Types Wizard, follow the steps below:

Step	Description	See Also																				
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Code Modules checkbox. The MDG Technology Wizard - Code Modules dialog displays.	Creating MDG Technologies, Step 6 [1065]																				
2	Click on the checkboxes (Product , Data Types , and Code Templates) for each of the required Code Modules. The code modules listed are those defined in your current project. These could be the Enterprise Architect default languages, or those you have defined yourself using code templates and the Code Template Editor. Before you can set up a code template for the new language in the editor, you must define at least one data type for the language. Once the MDG Technology file is created it can be loaded into your current model and into other models.	Code Templates Overview [1482] Code Template Editor [1485] Data Types [777]																				
3	To select any code options for a module, click on the (...) button in the Code Options column for that module. This enables you to select an XML document that provides additional settings for the language that are not covered by the data types or code templates. The root node of the XML document should be <code>CodeOptions</code> . The child nodes should be called <code>CodeOption</code> and should contain a <code>name</code> attribute. The supported code options are as follows:	Field Substitution Macros [1116]																				
	<table border="1"> <thead> <tr> <th>Code Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ConstructorName</td> <td>The name of a function used as a constructor for the <code>classHasConstructor</code> code template macro.</td> </tr> <tr> <td>CopyConstructorName</td> <td>The name of a function used as a copy constructor for the <code>classHasCopyConstructor</code> code template macro.</td> </tr> <tr> <td>DefaultExtension</td> <td>The default extension when generating code.</td> </tr> <tr> <td>DefaultSourceDirectory</td> <td>The default path to which Enterprise Architect generates code.</td> </tr> <tr> <td>DestructorName</td> <td>The name of a function used as a destructor for the <code>classHasDestructor</code> code template macro.</td> </tr> <tr> <td>Editor</td> <td>The external editor used for editing source code.</td> </tr> <tr> <td>ImplementationExtension</td> <td>The extension used by Enterprise Architect for the implementation file.</td> </tr> <tr> <td>ImplementationPath</td> <td>The relative path from the source file to the implementation file.</td> </tr> <tr> <td>PackagePathSeparator</td> <td>The delimiter used to separate package names in the <code>packagePath</code> macro from the code template macro.</td> </tr> </tbody> </table>	Code Option	Description	ConstructorName	The name of a function used as a constructor for the <code>classHasConstructor</code> code template macro.	CopyConstructorName	The name of a function used as a copy constructor for the <code>classHasCopyConstructor</code> code template macro.	DefaultExtension	The default extension when generating code.	DefaultSourceDirectory	The default path to which Enterprise Architect generates code.	DestructorName	The name of a function used as a destructor for the <code>classHasDestructor</code> code template macro.	Editor	The external editor used for editing source code.	ImplementationExtension	The extension used by Enterprise Architect for the implementation file.	ImplementationPath	The relative path from the source file to the implementation file.	PackagePathSeparator	The delimiter used to separate package names in the <code>packagePath</code> macro from the code template macro.	
Code Option	Description																					
ConstructorName	The name of a function used as a constructor for the <code>classHasConstructor</code> code template macro.																					
CopyConstructorName	The name of a function used as a copy constructor for the <code>classHasCopyConstructor</code> code template macro.																					
DefaultExtension	The default extension when generating code.																					
DefaultSourceDirectory	The default path to which Enterprise Architect generates code.																					
DestructorName	The name of a function used as a destructor for the <code>classHasDestructor</code> code template macro.																					
Editor	The external editor used for editing source code.																					
ImplementationExtension	The extension used by Enterprise Architect for the implementation file.																					
ImplementationPath	The relative path from the source file to the implementation file.																					
PackagePathSeparator	The delimiter used to separate package names in the <code>packagePath</code> macro from the code template macro.																					
	<p>An example of a valid code options file is shown below.</p> <pre><CodeOptions> <CodeOption name="DefaultExtension"> ext </CodeOption></pre>																					

Step	Description	See Also
	<pre><CodeOption name=" Editor " >C: \ Windows\ notepad. exe</ CodeOption > </ CodeOptions ></pre>	
4	<p>Click on the Next button to proceed.</p> <p>You can edit the code option values for source code engineering and for each required language using the appropriate Language Options page of the Options dialog.</p>	Code Engineering Settings ^[1516] Language Options ^[1529]

6.10.1.2.1.8 Add MDA Transforms

When creating an MDG Technology file, you can include the MDA Transformations that have been modified in the model.

To use the Transform Modules section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	<p>Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the MDA Transforms checkbox. The MDG Technology Wizard - Transform Modules dialog displays.</p>	Creating MDG Technologies, Step 6 ^[1065]
2	<p>Click the checkbox against the template name of each required template that is present in the current model.</p>	
3	<p>Click on the Next button to proceed.</p>	

6.10.1.2.1.9 Add Images

When creating an MDG Technology file, you incorporate images to be used in all models in which the technology is *deployed*. These images must already be available in the model in which the technology is being *developed*; you can import the images into this model using the **Add New** button on the **Image Manager**.

To use the Image Selection section of the MDG Technology Wizard, follow the steps below

Step	Description	See Also
1	<p>Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Images checkbox. The MDG Technology Wizard - Image Selection dialog displays.</p>	Creating MDG Technologies, Step 6 ^[1065]
2	<p>For each required model image available in the current model, select the checkbox next to the image name. A preview of each image displays on the right of the dialog as you select the checkbox.</p>	
3	<p>Click on the Next button to proceed.</p>	

Learn More:

- [Image Manager](#)^[593]

6.10.1.2.1.10 Add Scripts

When creating an MDG Technology file, you can include scripts that you have created in the model.

To use the Script Selection section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Scripts checkbox. The MDG Technology Wizard - Scripts dialog displays.	Creating MDG Technologies, Step 6 <small>[1065]</small>
2	For each required script available in the current model, select the checkbox next to the script name.	
3	Click on the Next button to proceed.	

Notes:

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

Learn More:

- [The Scriptor Window](#)
[1817]

6.10.1.2.1.11 Add RTF Report Templates

When creating an MDG Technology file, you can include user-defined RTF Report templates.

To use the report templates section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the RTF Templates checkbox The MDG Technology Wizard - RTF Report Templates dialog displays	Creating MDG Technologies, Step 6 <small>[1065]</small>
2	For each required user-defined report template available in the current model, select the checkbox next to the template name	
3	Click on the Next button to proceed	

6.10.1.2.1.12 Add Linked Document Templates

When creating an MDG Technology file, you can include Linked Document templates.

To use the Linked Document templates section of the MDG Technology Wizard, follow the steps below:

Step	Description	See Also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the Linked Document Templates checkbox. The MDG Technology Wizard - Linked Document Templates dialog displays.	Creating MDG Technologies, Step 6 <small>[1065]</small>

Step	Description	See Also
2	For each required document template available in the current model, select the checkbox next to the template name.	
3	Click on the Next button to proceed.	

6.10.1.2.2 Working with MTS Files

An MDG Technology Selection (.MTS) file stores the selected options that you define when creating an MDG Technology File using the MDG Technology Wizard. If you use a .MTS file, you can edit it to change the features selected when you generated the file, and to add or remove the advanced features described in this topic.

Action	Description	See also
Create an .MTS File	To create a .MTS file, select the Tools Generate MDG Technology File menu option to launch the MDG Technology Wizard, and work through the screens as described in the Create MDG Technology topic. On the second page, select the Create a new MTS file option.	Create MDG Technologies ^[1064]
Advanced Options For Your .MTS File	<p>Once you have created the .MTS file, you can add:</p> <ul style="list-style-type: none"> Model Search definitions. If you use a custom SQL search, the SQL must include <i>ea_guid AS CLASSGUID</i> and the <i>object type</i>. Model Views. Technology views do not store Favorite packages, only Views. If your exported views run searches that you have defined you must also include those searches in your MDG Technology. Model Validation configurations Model Templates. <p>Open the .MTS file in a text editor. To make it easier for you, you can copy the following lines and paste them into the file before the last line of the file (that is, just before the <code></ MDG. Sel ect i ons ></code> lines:</p> <pre><Model Searches file=" " / > <Model Views file=" " / ></pre> <p>(The code for the model validation configurations and model templates is provided in the corresponding sections, accessed via the links in the list above.)</p> <p>You can, if necessary, have more than one line for each inclusion; for example, more than one <i>ModelSearch</i>. For each inserted line:</p> <ul style="list-style-type: none"> In the <code>f i l e</code> attribute, enter the filename of the Model Search XML file or Model View XML file. <p>Save the .MTS file.</p>	Model Search Definitions ^[482] Model Views ^[469] Model Validation Configurations ^[1086] Model Templates ^[1087]
Update the MDG Technology	Again select the Tools Generate MDG Technology File menu option, but this time on the second page select Open an Existing MTS file and specify the file path of the .MTS file you have updated. Click on <Next> until the wizard is finished. Your MDG Technology file is updated.	

6.10.1.2.3 Customize Toolbox Profiles

The following is a road map of how to create a set of custom toolboxes for Enterprise Architect.

Step	Action	See also
1	Create a set of Toolbox Profiles that contain the definitions that Enterprise Architect requires to create the toolboxes.	Toolbox Profiles ^[1074]
2	Create a .MTS file containing instructions on how to build your MDG Technology. Use this .MTS file to build your MDG Technology.	Creating MDG Technologies ^[1064]
3	Add some finishing touches: <ul style="list-style-type: none"> • Create hidden sub-menus • Override Enterprise Architect's default toolboxes • Change the default icons for toolbox items. 	Hidden Sub-menus ^[1075] Overriding Default Toolboxes ^[1076] Icons for Toolbox Items ^[1076]

6.10.1.2.3.1 Create Toolbox Profiles

You can create multiple toolbox profiles within an MDG Technology. Each toolbox profile contains definitions that determine what appears in the **Toolbox** when a specific **Toolbox** page is open, either by selecting from the **More tools...** option in the **Toolbox** window, or by opening or creating a diagram of the type that is linked to the toolbox profile.

To create a toolbox profile, follow the steps below

Step	Action	See also
1	Create a diagram in a profile package, and give it a name by which you can refer to it later, such as <i>MyClassDiagram</i>	
	In the Notes field, give the diagram an alias and a description in the following format: <pre>Alias=MyClass; Notes=Structural elements for class diagrams;</pre>	
2	On the diagram, create a Class, name it <i>ToolboxPage</i> and give it the «metaclass» stereotype	
3	Create a «stereotype» element for each of the toolbox pages to create within your toolbox, such as <i>MyClassElements</i> and <i>MyClassRelationships</i> Set their Alias to the text to display in the title bar of each toolbox page, such as <i>My Class Elements</i> and <i>My Class Relationships</i> respectively Use the Notes field to define the tool-tip for each toolbox page; that is, Elements for Class Diagrams and Relationships for Class Diagrams Use the «extends» connector to set the stereotype elements to extend <i>ToolboxPage</i>	Toolbox Page Attributes ^[1075]
4	In the «stereotype» elements, create an attribute for each toolbox item The name of the attribute should be the name of the element or connector to be dropped, including namespace; for example, <i>UML::Package</i> , <i>UML::Class</i> and <i>UML::Interface</i>	

Step	Action	See also
	<p>The toolbox items display in the same order as the attributes in the Class, so make use of the attribute ordering buttons to define the order of your toolbox</p> <p>To name an attribute for an item from your own technology, precede it with your profile name as the namespace, and then follow it in brackets with the element or connector type that you are extending (so that Enterprise Architect knows what object to create); for example, a SysML block element would appear as <code>SysML::Block(UML::Class)</code></p> <p>Click on the links on the right for a complete list of elements and connectors that can be extended</p> <p>To define a toolbox item that allows a pattern to be dropped onto a diagram, name the attribute <code>My Technology::MyPattern(UMLPattern)</code> where <code>MyTechnology</code> is the ID of the technology and <code>MyPattern</code> is the name of the pattern to drop; for example, <code>BusFramework::Builder(UMLPattern)</code></p> <p>You can replace <code>(UMLPattern)</code> with <code>(UMLPatternSilent)</code> to avoid displaying the Add Pattern dialog</p> <p>To define a model-based pattern in a custom toolbox (such as the GoF patterns) create an attribute with a name of the format <code>PatternCategory::PatternName(UMLPattern)</code>; for example: <code>GoF Behavioral Patterns:: Mediator(UML Pattern)</code></p> <p>You might not want to use names such as <code>UML::Package</code> or <code>UML::Class</code> in your toolbox, so give the attributes an Initial Value of, for example, <code>Package</code> or <code>Class</code></p>	<p>Elements Used in Toolboxes ^[1077]</p> <p>Connectors Used in Toolboxes ^[1078]</p>
5	To save the toolbox profile, right-click on the diagram and select the Save as Profile context menu option	

Notes:

- Each profile element incorporated into an MDG Toolbox page enables synchronization of the Tagged Values and Constraints of all elements created from them

Learn More:

- [Synchronize Tags and Constraints](#) ^[1029]

The following attributes can be added to a stereotype Class that extends the `ToolboxPage` metaclass:

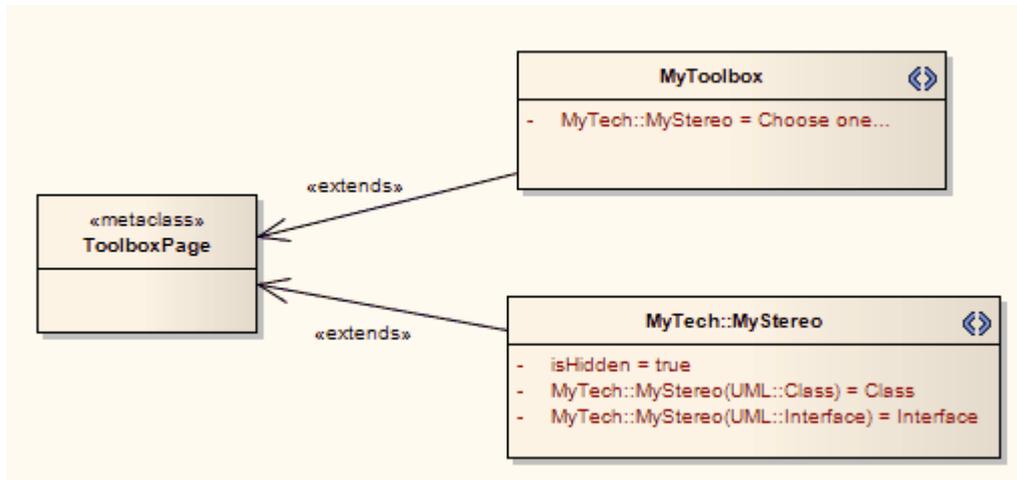
- **ImagesOnly:** if you give a toolbox page an attribute named `imagesOnly` with **Initial Value** set to **true**, the toolbox page displays without the text labels next to the icons
- **isCommon:** if you give a toolbox page an attribute named `IsCommon` with **Initial Value** set to **true**, the toolbox page is common to all defined toolboxes while your technology is active; the page is initially displayed as collapsed
- **isCollapsed:** if you give a toolbox page an attribute named `IsCollapsed` with **Initial Value** set to **true**, the toolbox page is initially minimized.
- **Icon:** see [Assign Icons for Toolbox Items](#) ^[1076]
- **isHidden:** see [Create Hidden Sub-Menus](#) ^[1075]

6.10.1.2.3.2 Create Hidden Sub-Menus

To create a sub-menu, create an additional «*stereotype*» element in the same toolbox profile and give it an attribute named `isHidden` with **Initial Value** of **true**. Define the toolbox item attributes as before. In the parent «*stereotype*» element, create an attribute with the identical name to the sub-menu element. The sub-menu element can have an alias.

This technique is very useful for 'disambiguating' stereotypes that can be applied to multiple metaclasses. In the example below, the «*MyStereo*» stereotype can be applied to either a Class or an Interface. On dragging

and dropping one from the toolbox, a hidden menu displays giving the choice of Class or Interface, then the appropriate element is dropped:



6.10.1.2.3.3 Override Default Toolboxes

Enterprise Architect has many default Toolbox Profiles, one for each of its inbuilt diagram types. These define the Toolbox pages that are displayed, by default, every time a diagram of a specific type is opened or brought into view.

To replace one of Enterprise Architect's default toolboxes with one of your own (for example, if you have your own version of the UML::Class toolbox that you want to be opened every time a Class diagram is opened - as long as your technology is active) then include a *RedefinedToolbox* clause in the **Notes** field for the diagram properties of your Toolbox Profile diagram. For example, the profile diagram's **Notes** field could resemble the following:

```
RedefinedToolbox=UML::Class; Alias=Class; Notes=Structural elements for Class diagrams;
```

This states that the toolbox defined by this profile replaces the Enterprise Architect toolbox *UML::Class* as the default toolbox for all UML Class diagrams. For a list of inbuilt toolboxes, see the [List of Enterprise Architect Toolboxes](#)^[1077] topic.

6.10.1.2.3.4 Assign Icons To Toolbox Items

To assign an icon to a toolbox item, follow the steps below

Step	Action	See also
1	Create a new «stereotype» element in the same toolbox profile as the toolbox item.	
2	Give the «stereotype» element the same name as the attribute that it is assigning an image to (for example, MyTech::MyStereo(UML::Class) in the diagram below)	
3	Give the «stereotype» element an attribute named <i>Icon</i> with Initial Value set to the full path and file name of the image to be used. The icon image is a 16x16 pixel bitmap file; for a transparent background, use light grey - RGB(192,192,192).	

Step	Action	See also
4	Create an Extends association from the «stereotype» element to a «metaclass» element named <i>ToolboxItemImage</i> .	



6.10.1.2.3.5 List of Enterprise Architect Toolboxes

The following is a list of the **Toolbox** pages that can be overridden:

- UML::Activity
- UML::Class
- UML::Communication
- UML::Component
- UML::Composite
- UML::Deployment
- UML::Interaction
- UML::Metamodel
- UML::Object
- UML::Profile
- UML::State
- UML::Timing
- UML::UseCase
- Extended::Analysis
- Extended::Custom
- Extended::DataModeling
- Extended::Maintenance
- Extended::Requirements
- Extended::UserInterface
- Extended::WSDL
- Extended::XMLSchema

6.10.1.2.3.6 Elements Used in Toolboxes

The following elements (all preceded with the namespace **UML::**) can be extended or redefined in Enterprise Architect Toolbox pages. The text in red indicates the label name displayed in the default Enterprise Architect Toolbox pages, where this differs in any way from the **UML::** statement text.

When these profile elements are incorporated into an MDG Toolbox page, they enable [synchronization](#)^[1029] of the Tagged Values and Constraints of all elements created from them.

You can also extend [connectors](#)^[1079].

- Action
- InteractionState (**State/Continuation**)

- Activity
- ActivityFinal (**Final**)
- ActivityInitial (**Initial**)
- ActivityParameter
- ActivityPartition (**Partition**)
- ActivityRegion (**Region**)
- Actor
- Artifact
- AssociationElement (**Association**)
- Boundary (for Use Cases)
- CentralBufferNode (**Central Buffer Node**)
- Change
- Choice
- Class
- Collaboration
- CollaborationOccurrence (**Collaboration Use**)
- Comment (**Note**)
- Component
- Constraint
- Datastore
- Decision
- DeploymentSpecification (**Deployment Specification**)
- Device
- DiagramLegend (**Diagram Legend**)
- DiagramNotes (**Diagram Notes**)
- DocumentArtifact (**Document Artifact** or **Document**)
- Entity (**Information**)
- EntityObject (**Entity**)
- EntryState (**Entry**)
- Enumeration
- ExceptionHandler (**Exception**)
- ExecutionEnvironment (**Execution Environment**)
- ExitState (**Exit**)
- Feature
- FinalState (**Final**)
- FlowFinalNode (**Flow Final**)
- ForkJoinH (**Fork/Join** - Horizontal)
- ForkJoinV (**Fork/Join** - Vertical)
- Interface
- Issue
- Junction
- Lifeline
- MergeNode (**Merge**)
- MessageEndPoint (**Endpoint** or **Message Endpoint**)
- MessageLabel (**Message Label**)
- Metaclass
- Node
- Object
- ObjectBoundary (**Boundary**)
- ObjectControl (**Control**)
- ObjectEntity (**Entity**)
- Package
- PackagingComponent
- Part
- Port
- Primitive
- Process
- Profile
- ProvidedInterface (**Expose Interface**)
- ReceiveEvent (**Receive**)
- Requirement
- RobustBoundary (**Boundary**)
- RobustControl (**Control**)
- RobustEntity (**Entity**)
- Screen
- SendEvent (**Send**)
- SequenceBoundary (**Boundary**)
- SequenceControl (**Control**)
- SequenceEntity (**Entity**)
- Signal
- State
- StateMachine (**State Machine**)
- StateTimeLine (**State Lifeline**)
- Stereotype
- StructuredActivity (**Structured Activity**)
- SynchState (**Synch**)

- Gate (**Diagram Gate**)
- GUIElement (**UI Control**)
- HistoryState (**History**)
- Hyperlink
- InformationItem (**Information Item**)
- InitialState (**Initial**)
- InteractionFragment (**Fragment**)
- Table
- Terminate
- TestCase (**Test Case**)
- Text
- UseCase (**Use Case**)
- UMLBoundary (**Boundary**)
- ValueTimeLine (**Value Lifeline**)

6.10.1.2.3.7 Connectors Used In Toolboxes

The following connectors (all preceded with the namespace **UML::**) can be extended or redefined in Enterprise Architect toolboxes. The text in red indicates the label name displayed in the default Enterprise Architect Toolbox pages, where this differs in any way from the **UML::** statement text.

You can also extend [elements](#)^[1077].

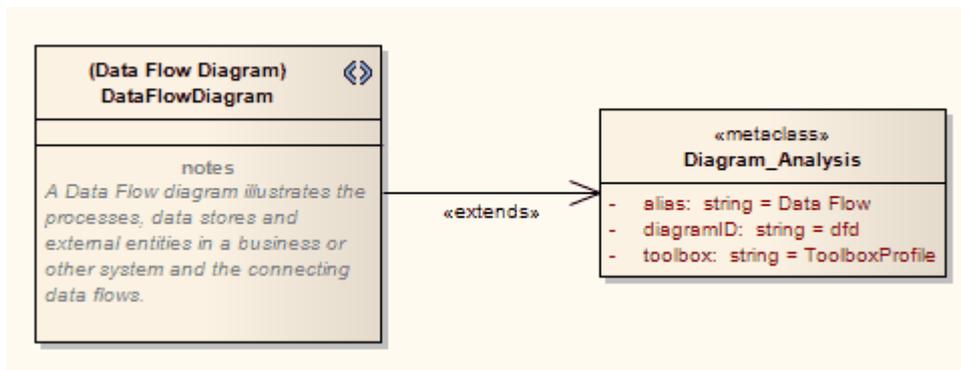
- Aggregation (**Aggregate**)
- Assembly
- Association (**Associate**)
- AssociationClass (**Association Class**)
- CallFromRecursion (**Call**)
- CommunicationPath (**Communication Path**)
- Composition (**Compose**)
- Connector
- ControlFlow (**Control Flow**)
- Delegate
- Dependency
- Deployment
- Extension
- Generalization (**Generalize** or **Inheritance**)
- InformationFlow (**Information Flow**)
- InterruptFlow (**Interrupt Flow**)
- Invokes
- Manifest
- Message
- Nesting
- NoteLink (**Note Link**)
- ObjectFlow (**Object Flow**)
- Occurrence
- PackageImport (**Package Import**)
- PackageMerge (**Package Merge**)
- Precedes
- ProfileApplication (**Application**)
- Realization (**Realize** or **Implements**)
- Recursion
- Redefinition
- Representation
- Represents
- RoleBinding (**Role Binding**)
- SelfMessage (**Self-Message**)
- TagValAssociation (**Tagged Value**)
- TemplateBinding (**Template Binding**)
- TraceLink (**Trace**)
- Transition
- UCExtend (**Extend**)
- UCInclude (**Include**)
- UseCaseLink (**Use**)

6.10.1.2.4 Create Custom Diagram Profiles

You can create extended diagram types in Enterprise Architect and include them in MDG Technologies. To create extended diagram types, follow the steps below:

Step	Action	See also
1	Create a profile with the same name as the MDG Technology in which it is to be included; for example, SysML. (However, check the Note in <i>Define Child Diagram Types</i> .)	Defining Child Diagram Types ^[1057]
2	Create a « <i>stereotype</i> » Class element that is named as the custom diagram, for example, <i>BlockDefinition</i> .	
3	Create a Class element and name it as one of the Built-In Diagram Types prefixed with <i>Diagram_</i> , for example <i>Diagram_Logical</i> for Class diagrams or <i>Diagram_Use Case</i> for Use Case diagrams.	Built-In Diagram Types ^[1081]
4	Give the <i>Diagram_x</i> Class the « <i>metaclass</i> » stereotype and draw an « <i>extends</i> » connector from the stereotype to the metaclass.	
5	In the Notes field, give the stereotype Class a brief description of what the diagram is used for. This description displays in the bottom right-hand corner of the New Diagram dialog.	
6	Give the <i>Diagram_x</i> Class the following attributes as required: <ul style="list-style-type: none"> • <i>alias</i>: <i>string</i> = <i>Type</i> (where <i>Type</i> appears before the word 'Diagram' on the diagram title bar) • <i>diagramID</i>: <i>string</i> = <i>abc</i> (where <i>abc</i> is the diagram type that appears in the diagram frame label) • <i>toolbox</i>: <i>string</i> = <i>ToolboxName</i> (where <i>ToolboxName</i> is the name of the toolbox profile for the toolbox that opens automatically each time a diagram is opened) • <i>frameString</i>: <i>string</i> = <i>FrameFormatString</i> (where <i>FrameFormatString</i> is a string containing substitution macros for defining the frame title, with or without additional delimiters such as () ; macros that can be used are: <ul style="list-style-type: none"> • #DGMSTEREO# • #DGMID# • #DGMTYPE# • #DGMALIAS# • #DGMOWNERNAME# • #DGMOWNERNAMEFULL# • #DGMNAME# • #DGMNAMEFULL# • <i>swimlanes</i>: <i>string</i> = <i>Lanes=2;Orientation=Horizontal;Lane 1=Title 1; Lane 2=Title 2;</i> (where <i>Lanes</i> can be any value, but the number of <i>Lane<n></i> values must equal the value of <i>Lanes</i>; <i>Orientation</i> can be omitted, in which case the swimlanes default to vertical) • <i>StyleEx</i>: <i>string</i> = one or more of a range of values; see Attribute Values - StyleEx & pdata • <i>pdata</i>: <i>string</i> = one or more of a range of values; see Attribute Values - StyleEx & pdata. 	Diagram Frame ^[888] Attribute Values - StyleEx & pdata ^[1081]
7	Save the diagram as a profile in the usual manner.	
8	Add the diagram profile to the .MTS file used in the MDG Technology.	Add Diagram Types ^[1088]

The following example shows the DFD diagram profile which defines a DFD diagram as an extension of the Enterprise Architect Analysis diagram.



6.10.1.2.4.1 Built-In Diagram Types

The following is a full list of built-in diagram types provided by Enterprise Architect.

- Activity
- Analysis
- Collaboration
- Component
- CompositeStructure
- Custom
- Deployment
- InteractionOverview
- Logical
- Object
- Package
- Sequence
- Statechart
- Timing
- Use Case

Note the use of *Logical* for Class diagrams and also notice the space in the middle of *Use Case*. These names are used in [Defining Child Diagram Types](#)^[1057], or prefixed by *Diagram_* in creating [Diagram Profiles](#)^[1080].

6.10.1.2.4.2 Attribute Values - StyleEx & pdata

When creating a diagram profile, you can use the *pdata* and *StyleEx* attributes to define a range of characteristics of the diagrams created with the profile. If the attribute is defining several characteristics at once, put the values in a single string separated by semicolons. For example:

StyleEx: string = HideQuals=0;AdvanceElementProps=1;ShowNotes=1;

StyleEx: string =

- *AdvancedConnectorProps=1;* (to show connector property strings)
- *AdvancedElementProps=1;* (to show the element property string)
- *AdvancedFeatureProps=1;* (to show the feature property string)
- *AttPkg=1;* (to show package visible Class members)
- *DefaultLang=Language;* (to set the default language for the diagram; *Language* can be one of Enterprise Architect's built-in languages such as C++ or Java, or it can be a custom language)
- *HandDrawn=1;* (to apply hand drawn mode)
- *HideQuals=0;* (to show qualifiers and visibility indicators)

- *SeqTopMargin=50*; (to set the height of the top margin on sequence diagrams)
- *ShowAsList=1*; (to make the diagram open directly into the [Diagram List](#)^[462])
- *ShowMaint=1*; (to show the element Maintenance compartment)
- *ShowNotes=1*; (to show the element Notes compartment)
- *ShowOpRetType=1*; (to show the operation return type)
- *ShowTests=1*; (to show the element Testing compartment)
- *SuppConnectorLabels=1*; (to suppress all connector labels)
- *SuppressBrackets=1*; (to suppress brackets on operations without parameters)
- *TConnectorNotation=Option*; (where *Option* is one of **UML 2.1**, **IDEF1X**, or **Information Engineering**)
- *TExplicitNavigability=1*; (to show non-navigable connector ends)
- *VisibleAttributeDetail=1*; (to show attribute details on the diagram)
- *Whiteboard=1*; (to apply whiteboard mode)

pdata: string =

- *HideAtts=0*; (to show the element Attributes compartment)
- *HideEStereo=0*; (to show element stereotypes in the diagram)
- *HideOps=0*; (to show the element Operations compartment)
- *HideParents=0*; (to show additional parents of elements in the diagram)
- *HideProps=0*; (to show property methods)
- *HideRel=0*; (to show relationships)
- *HideStereo=0*; (to show attribute and operation stereotypes)
- *OpParams=3*; (to show operation parameters)
- *ShowCons=1*; (to show the element Constraints compartment)
- *ShowIcons=1*; (to use stereotype icons)
- *ShowReqs=1*; (to show the element Requirements compartment)
- *ShowSN=1*; (to show sequence notes)
- *ShowTags=1*; (to show the element Tagged values compartment)
- *SuppCN=0*; (to show collaboration numbers)
- *UseAlias=1*; (to use the aliases or elements in the diagram, if available).

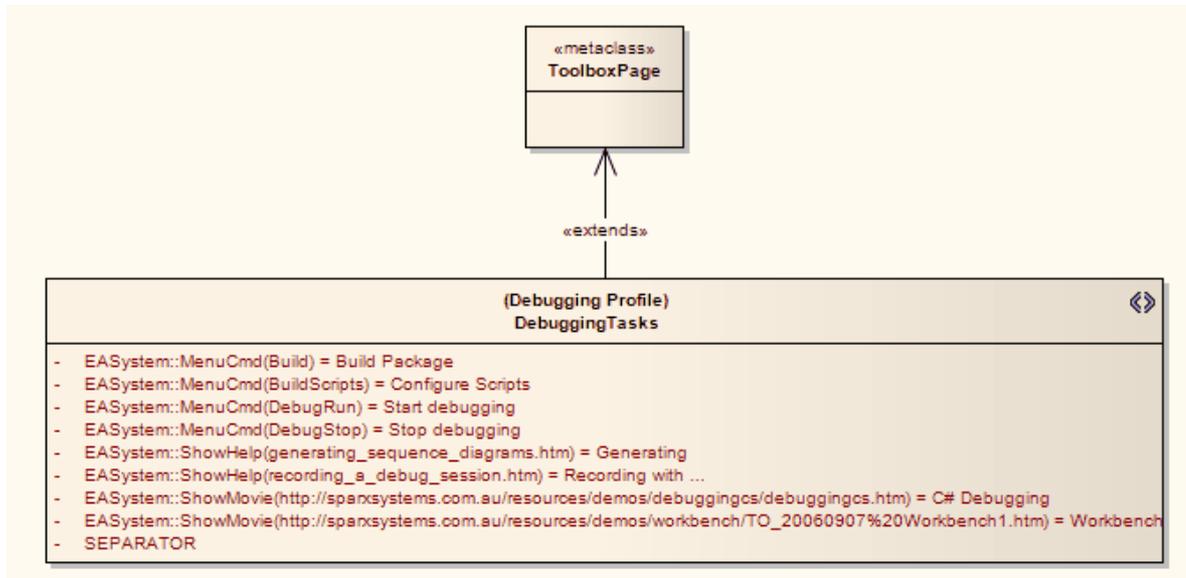
6.10.1.2.5 Create Learning Center Profiles

Defining Learning Center profiles is a four-part process:

Step	Action	See also
1	Define toolboxes; create any number of stereotype elements that each define a Learning Center Toolbox page	Define Learning Center Toolboxes ^[1083]
2	Define contexts; create any number of stereotype elements that each define a named context A context might be when a specific diagram type is open, or when a specific element type is selected	Define Learning Center Contexts ^[1085]
3	Allocate contexts to toolboxes; define the many-to-many relationships between the Learning Center toolboxes and the available contexts	Allocate Learning Center Contexts ^[1086]
4	Create the profile and incorporate it into your Technology	Create Learning Center Profile ^[1086] Add Learning Center Pages ^[1069]

6.10.1.2.5.1 Define Learning Center Toolboxes

A Learning Center toolbox is defined by a «stereotype» Class that extends a «metaclass» ToolboxPage element. These elements must be owned by a «profile» package. Each «stereotype» Class represents the contents of the Learning Center for a given context, and each attribute of the «stereotype» Class defines a command button in the Learning Center. The following diagram shows an example of a Learning Center toolbox.



The title bar of the Learning Center toolbox is defined by the Alias of the «stereotype» Class, in this case *Debugging Profile*. This example uses the following standard attribute types:

- **EASystem::MenuCmd.** These entries name an Enterprise Architect main menu command inside round brackets. See the complete [list of inbuilt commands](#) ^[1084]. Type the text to appear in the Learning Center into the **Initial Value** field.
- **EASystem::ShowHelp.** These entries name a page from the *Enterprise Architect User Guide* inside round brackets. To find out the names of pages in the *Enterprise Architect User Guide*, right-click on the page and select the **Properties** context menu option. Type the text to appear in the Learning Center into the **Initial Value** field.
- **EASystem::ShowMovie.** These entries give the URL of a movie inside round brackets. Type the text to appear in the Learning Center into the **Initial Value** field.
- **SEPARATOR.** This entry indicates that a separator should be placed in the Learning Center toolbox. If it is necessary to place multiple separators in a single toolbox, note that Enterprise Architect does not allow identically named attributes for a Class: simply change the case of one or more letters to get around the problem.

Other useful attributes include:

- **EASystem::ShowURL.** This entry gives the URL of a web page inside round brackets. Type the text to appear in the Learning Center into the **Initial Value** field.
- **isCommon:** A boolean attribute with **Initial Value** set to **True**, defines a Learning Center toolbox as context-free and common, appearing for all contexts
- You can also [run Add-In functions](#) ^[1084] from the Learning Center.

Next Step

The next step is to create a set of [Learning Center Contexts](#) ^[1085].

The following Enterprise Architect commands can all be used in user-defined Learning Center profiles. Learning Center pages have attributes named in the form *EASystem::MenuCmd(<CommandName>)* where *<CommandName>* is the name chosen from the following list:

- AddDiagram
- AddElement
- AddPackage
- AutoRecordThread
- AddModelFromPattern
- Build
- BuildScripts
- ConfigureCSV
- ConfigureValidation
- CreateBaseLine
- CreateSequenceDiagram
- DebugPause
- DebugRun
- DebugStop
- Deploy
- DiagramsOnlyReport
- ElementUsage
- ExportXML
- FileNew
- FileOpen
- GenerateDDL
- GenerateWSDL
- GenerateXMLSchema
- ImplementationDetails
- ImportBinary
- ImportExportCSV
- ImportSchema
- ImportSourceDirectory
- ImportWSDL
- ImportXML
- ImportXMLSchema
- Run
- RunHTMLReport
- RunRTFReport
- SetClassifier
- ShowHideExecution
- StartDebugRecording
- StepInto
- StepOut
- StepOver
- StopDebugRecording
- Test
- TestingReport
- ToggleLevelNumbering
- TransformPackage
- TransformSelectedElements
- ValidateModel
- ViewAuditing
- ViewDebug
- ViewElementList
- ViewForum (for **Team Review** window)
- ViewHierarchy (for **Traceability** window)
- ViewMaintenance
- ViewOutput
- ViewProjectManagement
- ViewRelationships
- ViewRelMatrix
- ViewRequirementTypes
- ViewRules
- ViewSearch
- ViewSourceCode
- ViewTaggedValues
- ViewTesting
- ViewTestingDetails
- ViewWebBrowser

To run Add-In functions from the Learning Center, you create an attribute in the Learning Center «*stereotype*» Class with the following format:

```
" Assembly : : FunctionName ( ) "
```

where *Assembly* is the name of the Add-In and *FunctionName* is the name of a public function in the Add-In. Give the attribute an initial value of the text that is to appear in the Learning Center. The function receives two parameters and returns a success status, as in the following VB.Net example:

```
Public Function ShowMyDiagram(ByRef Repository As EA.Repository, ByVal
args As Object) As String
    Dim ret As String
    ret = Repository.SQLQuery("select ea_guid from t_diagram where
diagram_type='Custom' and StyleEx like
'*;MDGDgm=MyDiagrams::MyCustomDiagram;*'")
    If ret Is Nothing Then
        ShowMyDiagram = False
        Exit Function
    End If

    Dim oXML As MSXML2.DOMDocument = New MSXML2.DOMDocument
    oXML.LoadXML(ret)
```

```

Dim NodeList As MSXML2.IXMLDOMNodeList = oXML.selectNodes("//
ea_guid")
If NodeList.Length = 0 Then
    ShowMyDiagram = False
    Exit Function
End If

Dim Node As MSXML2.IXMLDOMNode
Dim diag As EA.Diagram
If NodeList.Length >= 1 Then
    Node = NodeList.item(0)
    diag = Repository.GetDiagramByGuid(Node.text)
    Repository.OpenDiagram(diag.DiagramID)
    Repository.ShownProjectView(diag)
End If

ShowMyDiagram = True
End Function

```

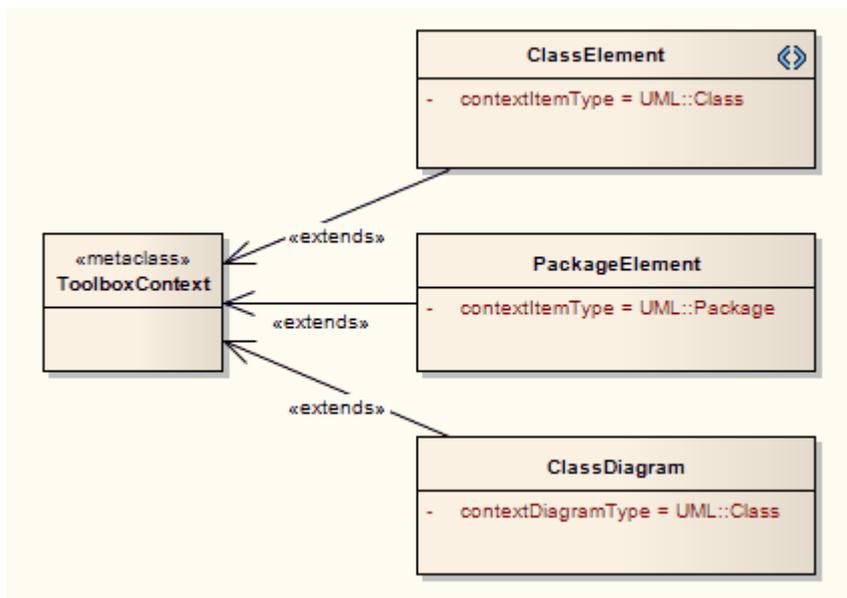
6.10.1.2.5.2 Define Learning Center Contexts

Named Contexts

To create a context-sensitive set of Learning Centers, you must define a set of named contexts that can be used in the definition. A named context is a «*stereotype*» Class which extends a «*metaclass*» named *ToolboxContext*. These elements must be owned by the same «*profile*» package that owns the [Learning Center toolbox definitions](#) ^[1085]. The context Class has one of the following attributes:

- contextDiagramType; this should have an **Initial Value** set to a valid diagram type
- contextItemType; this should have an **Initial Value** set to a valid element type
- contextKey.

Example Context Profile

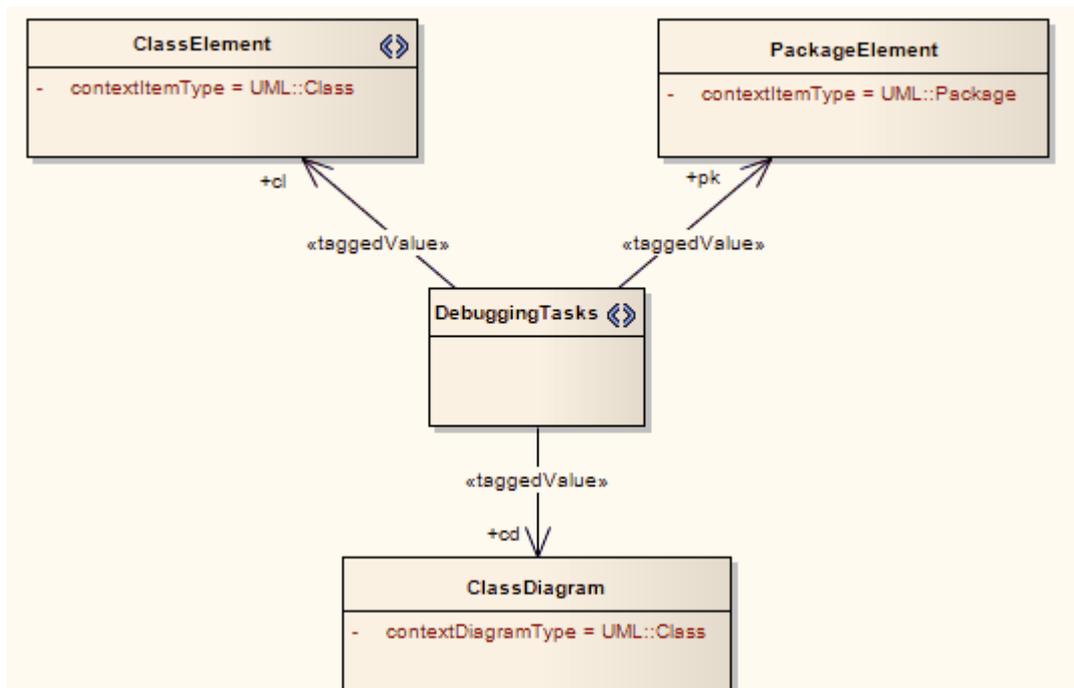


Next Step

The next step is to [allocate Learning Center contexts to Learning Center toolboxes](#) ^[1086].

6.10.1.2.5.3 Allocate Learning Center Contexts

Once you have defined your [Learning Center toolboxes](#) ^[1083] and [Learning Center contexts](#) ^[1085], you can allocate your toolboxes as many contexts as apply, and any number of toolboxes can be allocated to a single context. This is done by creating a «taggedValue» connector from the toolbox element to the context element. The Association end at the context end must be named. The following diagram shows how this might appear.



Next Step

The next step is to [Save your Learning Center Profile](#) ^[1086].

6.10.1.2.5.4 Save a Learning Center Profile

The best organization structure for the model in which you are creating your Learning Center Profile is:

- A single «profile» package
- Three diagrams within the «profile» package named *Toolboxes*, *Contexts* and *Context Allocations*
- Each toolbox page «stereotype» element is owned by the «profile» package and appears on the *Toolboxes* and *Context Allocations* diagrams
- Each context «stereotype» element is owned by the «profile» package and appears on the *Contexts* and *Context Allocations* diagrams
- Each «metaclass» element is owned by the «profile» package and appears on the *Toolboxes* or *Contexts* diagram.

From this structure, creating a Learning Center Profile is as simple as right-clicking on the «profile» package in the Project Browser and selecting the **Save Package as UML Profile** context menu option.

6.10.1.2.6 Define Validation Configuration

The Model Validation Configuration dialog can be opened using the **Project | Model Validation | Configure...** menu option. Using this dialog, you can choose which sets of validation rules are and are not executed when a user performs a validation. Rather than perform this configuration manually and potentially have to change the settings every time Enterprise Architect is started and a different technology is set active, you can

define the configuration settings within the MTS file.

To specify a set of rules as a white-list (that is, anything added to this list is turned ON), open your MTS file in a text editor and copy and paste the following `<Model Validation>` block at the top level inside the `<MDG. Selections>` block:

```
<Model Validation>
  <RuleSet name="BPMNRules"/> <!-- ruleset ID defined in the Project.
  DefineRuleCategory call -->
  <RuleSet name="MVR7F0001"/> <!-- notice you can turn on/off system
  rules as well! -->
</Model Validation>
```

Ensure that the ruleset IDs do not contain any spaces.

To specify a set of rules as a black-list (that is, anything added to this list is turned OFF), open your MTS file in a text editor and copy and paste the following `<Model Validation>` block at the top level inside the `<MDG. Selections>` block:

```
<Model Validation isBlackList="true">
  <RuleSet name="BPMNRules"/>
  <RuleSet name="MVR7F0001"/>
</Model Validation>
```

In the examples above, "BPMNRules" is the rule-set ID defined in the Project.

DefineRuleCategory call - see [Project Interface](#)^[192] for details. "MVR7F0001" is one of Enterprise Architect's built-in rule-sets. These validation options are applied when you activate the appropriate technology. The global (default) technology has all rules turned on.

6.10.1.2.7 Incorporate Model Templates

Model Templates can be added into a model, either on creation of the model, or at any time by right-clicking on a package in the **Project Browser** and selecting the **Add | Add a New Model using Wizard...** context menu option. To include your own model templates in this dialog, add them to your MDG Technology as described in the following table:

Step	Action	See also
1	Create a package that contains all sub-packages, diagrams, elements, notes and information links that you want to provide in the model template. See the <i>Model Templates</i> section and the EAExample.eap model for illustrations of what you might include, or create a model from a standard template and see what is generated. Your model template package must be self contained and not contain any dependencies or other links to elements outside the package.	Model Templates ^[519] Model Wizard ^[518]
2	Export your package to XML.	Export to XML ^[320]
3	Create a reference to the XML file in the MTS file. Open your MTS file in a text editor and copy and paste the following <code><Model Templates></code> block at the top level inside the <code><MDG. Selections></code> block: <pre><Model Templates> <Model name="Template Name" description="This is the description." location="MyTemplatePackage.xml" default="yes" icon="34" filter="Filter Name"/> </Model Templates></pre>	Working with MTS Files ^[1073]

You can include as many `<Model Templates>` blocks in your MTS file as you have model templates. The attributes have the following meanings:

- **Model name:** The name of the model template to show in the **Select model(s)** dialog, which displays

when you create a new model or when you execute the **Add a New Model using Wizard** menu option.

- **description:** The text to display in the **Select model(s)** dialog when the name is selected.
- **location:** The path of the XML file that contains the XML export of the model template package, relative to the location of the MDG Technology file. If the XML file is in the same folder as the technology file then this just contains the file name.
- **default:** Contains either **yes** to indicate that the model template is checked by default, or **no** to indicate that the model template is un-checked by default.
- **icon:** Contains an index to Enterprise Architect's base icons list. To show the appropriate view icon, use one of the following values: **29** = Use Case, **30** = Dynamic; **31** = Class; **32** = Component; **33** = Deployment; **34** = Simple.
- **filter:** If you have a large number of model templates, you can group them on the **Select model(s)** dialog by giving all the model templates in the same group the same filter name. The filter name given appears in the **Select from:** list box in the **Select model(s)** dialog.

6.10.1.2.8 Set Up Technology Element Images

This topic outlines how to capture images to represent MDG Technology elements, so that the images are displayed on a diagram a user creates through the technology when it is deployed in the user's model. Note that you would probably work backwards and forwards through the steps many times, adding objects as you identify the requirement for them.

To capture images to represent MDG Technology elements, follow the steps below

Step	Action	See also
1	Import suitable images into the MDG Technology development model from various sources, using the Add New button on the Image Manager.	Image Manager ^[593]
2	Design and create a Stereotype (UML) Profile containing (if appropriate) a stereotype definition for each element or connector to be owned by the technology. These stereotype definitions can contain Shape Scripts that in turn incorporate the imported images.	Stereotype (UML) Profile ^[103] Shape Scripts ^[104] Drawing Methods ^[109]
3	Design and create a Toolbox Profile with stereotype elements that contain an attribute for each element or connector that can be dropped onto a diagram from the toolbox. These attributes identify the name of the technology element or connector, any modifying stereotype (which might incorporate the required image) and the UML or Extended element or connector on which the technology object is based. For example: <i>SysML::Block(UML::Class)</i> , where <i>SysML</i> is the Technology Profile, <i>UML::Class</i> is the UML element used as the base, and <i>Block</i> is the stereotype that modifies the Class to turn it into a SysML Block element.	Toolbox Profiles ^[107]
4	Design and create a Diagram Profile that identifies the Toolbox Profile. When a diagram of the type defined in the Diagram Profile is opened, it in turn opens a set of toolbox pages as defined by the Toolbox Profile.	Diagram Profiles ^[108]
5	Create or update the technology as required, adding the Stereotype Profile, Diagram Profile, Toolbox Profile and Image files to the technology from the development model.	Creating MDG Technologies ^[106] Adding Stereotype Profile ^[106] Adding Diagram Profile ^[106] Adding Toolbox Profile ^[106] Adding Image Files ^[107]

Step	Action	See also
6	Deploy the technology as appropriate. When a user applies the technology to their own model, and creates a diagram under that technology, the elements they create on the diagram should be represented by the images you assigned to those elements when you created the technology.	Deploying MDG Technologies ^[1089]

6.10.1.2.9 Deploy An MDG Technology

An MDG Technology can be deployed in one of two ways: as a file or from an Add-In.

Deploy From a File

To deploy your technology as a file, you have a number of choices:

- Import it into the %APPDATA% folder. See the [Import MDG Technologies](#) ^[1035] topic.
- Copy it to a folder named MDG Technologies, which you must create under your Enterprise Architect installation directory (by default this is C:\Program Files\Sparx Systems\EA). When you restart Enterprise Architect, your MDG Technology is deployed.
- Copy it to any folder in your file system, including network drives: use the Enterprise Architect **Settings | MDG Technologies...** menu option, press the **Advanced** button and add the folder to the Technologies path. This deployment method enables you to quickly and easily deploy a technology to all Enterprise Architect users on a LAN.
- Upload it to an internet or intranet location: use the Enterprise Architect **Settings | MDG Technologies...** menu option, press the **Advanced** button and add the URL to the Technologies path. This deployment method enables you to quickly and easily deploy a technology to an even wider group of Enterprise Architect users.

Deploy From an Add-in

To deploy your technology from an Add-In, you must write an [EA_OnInitializeTechnologies](#) ^[2035] function. The following example is written in VB.Net:

```
Public Function EA_OnInitializeTechnologies(ByVal Repository As EA.
Repository) As Object
    EA_OnInitializeTechnologies = My.Resources.MyTechnology
End Function
```

6.10.1.3 Shape Scripts

Introduction

Enterprise Architect *Shape Scripts* enable you to specify custom shapes via a scripting language. These custom shapes are drawn instead of the standard UML notation. Each script is associated with a particular stereotype, and is drawn for every element of that stereotype. The following topics describe how to create and apply Shape Scripts:

- [Getting Started with Shape Scripts](#) ^[1090]
- [Write Scripts](#) ^[1091]
- [Example Scripts](#) ^[1104]
- [Shape Editor](#) ^[1090]
- [Add Shape Scripts to UML Profiles](#) ^[1046]

6.10.1.3.1 Getting Started With Shape Scripts

Step	Action	See also
1	<i>Shape Scripts</i> are associated with stereotypes and are defined via the Stereotypes tab of the UML Types dialog. To access this dialog, select the Settings UML Types menu option. Each stereotype defined can have a Shape Script.	
2	You can create a Shape Script for an existing stereotype by selecting the stereotype from the list. Alternatively, you can create new stereotypes by clicking on the New button and giving the stereotype a name. Select a base Class and click on the Save button. Once the stereotype is saved, it displays in the list.	
3	To override the appearance, select the Shape Script radio button and then click on the Assign button. The Shape Script Editor displays.	
4	Type the example Shape Scripts in the Edit window. You can click on the Refresh button in order to view the shape in the preview window.	
5	If you define a composite Shape Script (such as the connector at the end of the Example Scripts topic), click on the Next Shape button to page through the components of the shape.	
6	Once you have finished writing your Shape Script, click on the OK button. To save the Shape Script you must click on the Save button on the Stereotypes tab.	
7	Once you have created your Shape Script for a particular stereotype, you can assign that stereotype to an element or connector. The appearance reflects the Shape Script you created. To do this, drag and drop the appropriate element or connector into your diagram.	
8	Right-click on the element or connector and select the Properties context menu option. Click on the Stereotype field drop-down arrow, select the stereotype you created and click on the OK button. The object's shape now reflects the Shape Script you created.	

Notes:

- If an element's appearance is modified by a Shape Script, many of the options on the **Advanced** context menu for that element are disabled.

Learn More:

- [Shape Script Editor](#)^[1090]
- [Writing Scripts](#)^[1091]
- [Example Scripts](#)^[1103]

6.10.1.3.2 Shape Editor

The **Shape Editor** enables you to create Shape Scripts. It provides the facilities of the *Common Code Editor*, including intellisense for Shape Script attributes and functions.

How to:

To access the Shape Editor, follow the steps below:

1. Select **Settings | UML Types > Stereotypes**
2. In the **Stereotype** field, type a name or click on an the required stereotype in the list
3. From the **Override Appearance** panel, select the **Shape Script** radio button
4. Click on the **Assign** button. The **Shape Editor** dialog displays

Complete the fields on the dialog as explained in the following table.

Field	Usage	See also
Format	Select the Shape Script version.	
Import	Import a Shape Script from a text file.	
Export	Export a Shape Script to a text file.	
OK	Exit from the Shape Editor , don't forget to save your script from the Stereotypes tab.	Getting Started With Shape Scripts ^[1090]
Next Shape	Rotate though the multiple shape definitions.	
Refresh	Parse your script and display the result in the Preview window.	

Learn More

- [Code Editors](#) ^[1394], for more information on intellisense and the Common Code Editor.

6.10.1.3.3 Write Scripts

This topic is a detailed reference for writing Shape Scripts. For an introduction to writing Shape Scripts, see the *Getting Started With Shape Scripts* and *Example Scripts* topics.

See the following reference topics for more detailed information on shape scripting:

- [Syntax Grammar](#) ^[1091]
- [Shape Attributes](#) ^[1092]
- [Drawing Methods](#) ^[1094]
- [Color Queries](#) ^[1099]
- [Conditional Branching](#) ^[1099]
- [Query Methods](#) ^[1099]
- [Display Element/Connector Properties](#) ^[1099]
- [Sub-shapes](#) ^[1102]
- [Reserved Names](#) ^[1103]
- [Miscellaneous](#) ^[1103]

Learn More:

- [Getting Started With Shape Scripts](#) ^[1090]
- [Example Scripts](#) ^[1104]

6.10.1.3.3.1 Syntax Grammar

Grammar symbols:

- * = zero or more
- + = one or more
- | = or
- ; = terminator

ShapeScript	::=	<Shape>*;
Shape	::=	<ShapeDeclaration> <ShapeBody>;
ShapeDeclaration	::=	<ShapeType> <ShapeName>;
ShapeType	::=	"shape" "decoration";
ShapeName	::=	<ReservedShapeName> <stringliteral>;
ReservedShapeName	::=	See Reserved Names ^[1093] for full reserved shape listing
ShapeBody	::=	"{" <InitialisationAttributeAssignment>* <DrawingStatement>* <SubShape>* "}";
InitialisationAttributeAssignment	::=	<Attribute> "=" <Value> ",";
Attribute	::=	See Shape Attributes ^[1092] for full listing of attribute names
DrawingStatement	::=	<IfElseSection> <Method>;
IfElseSection	::=	"if" "(" <QueryExpression> ")" <TrueSection> (<ElseSection>) ;
QueryExpression	::=	<QueryName> "(" <ParameterList> ")";
QueryName	::=	See Query Methods ^[1093] for a full listing of Query names
TrueSection	::=	"{" <DrawingStatement>* "}"
ElseSection	::=	"else" "{" <DrawingStatement>* "}"
Method	::=	<MethodName> "(" <ParameterList> ")" ",";
MethodName	::=	See Drawing Methods ^[1094] for a full listing of method names

6.10.1.3.3.2 Shape Attributes

syntax: attribute "=" value ","

example:

```

shape main
{
    // Initialisation attributes - must be before drawing commands
    noshadow = "true";
    h_align = "center";

    // drawing commands
    rectangle(0, 0, 100, 100);
    println("foo bar");
}

```

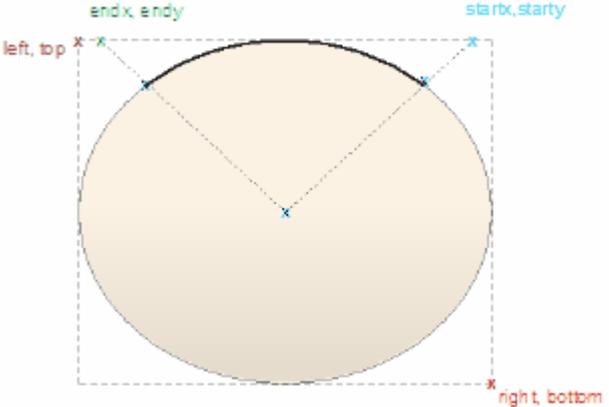
Attribute Name	Type	Description	See also
bottomAnchorOffset	(int,int)	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the bottom edge of its parent. For example: <i>bottomAnchorOffset=(0,-10)</i> ; move embedded element up 10 pixels from the bottom edge	

Attribute Name	Type	Description	See also
dockable	<i>string</i>	Makes the shape default to dockable, so that it can be aligned with and joined to other elements (both other Shape Scripts and standard elements) on a diagram. You cannot reverse the dockable status with the Appearance menu option; to change the status, you must edit the Shape Script. Valid values: standard or off	Appearance Menu Section ^[656]
editableField	<i>string</i>	Defines a shape as an editable region of the element. This field impacts element shapes only, line glyphs are not supported. Valid Values: alias , name , note , stereotype	
endPointY, endPointX	<i>integer</i>	Only used for the reserved target and source shapes for connectors; this point determines where the main connector line connects to the end shapes. Default: 0 and 0	
fixedAspectRatio	<i>string</i>	Set to true to fix the aspect ratio. Do not use if you do not want to fix the aspect ratio.	
h_Align	<i>string</i>	Affects horizontal placement of printed text and subshapes depending on the layoutType attribute. Valid values: left , center , or right	
layoutType	<i>string</i>	Determines how subshapes are sized and positioned. See Subshape Layout for further details. Valid values: leftright , topdown , border	Subshape Layout ^[1102]
leftAnchorOffset	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the left edge of its parent. For example: <i>leftAnchorOffset=(10,0)</i> ; move embedded element right 10 pixels from the left edge	
noShadow	<i>string</i>	Set to true to suppress the shapes shadow from being rendered. Valid values: true or false (default= false)	
orientation	<i>string</i>	Applies to decoration shapes only. Determines where the decoration is positioned within the containing element glyph. Valid values: NW , N , NE , E , SE , S , SW , W	
preferredHeight		Used by border layoutType - north and south Used in drawing the source and target shapes for connectors to determine how wide the line is.	
preferredWidth		Used by border layoutType - east and west. Used by leftright layoutType, shapes where	

Attribute Name	Type	Description	See also
		<i>scalable</i> is false to determine how much space they occupy for layout purposes.	
rightAnchorOffset	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the right edge of its parent. For example: <i>rightAnchorOffset=(-10,0)</i> ; move embedded element left 10 pixels from the right edge	
rotatable	<i>string</i>	Set to false to prevent rotation of the shape. This attribute is only applicable to the source and target shapes for lines glyphs. Valid values: true or false (default = true)	
scalable	<i>string</i>	Set to false to stop the shape from being relatively sized to the associated diagram glyph. Valid values: true or false (default= true)	
topAnchorOffset	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the top edge of its parent. For example: <i>topAnchorOffset=(0,10)</i> ; move embedded element down 10 pixels from the top edge	
v_Align	<i>string</i>	Affects vertical placement of printed text and subshapes depending on the layoutType attribute. Valid values: top , center , or bottom	

6.10.1.3.3.3 Drawing Methods

Method Name	Description	See also
addsubshape (string shapename (, int width, int height))	Adds a sub-shape with the name shapename that must be defined within the current shape definition	
arc (int left, int top, int right, int bottom, int startingpointx , int startingpointy , int endingpointx , int endingpointy)	Draws an elliptical anticlockwise arc with the ellipse having extents at left , top , right and bottom The start point of the arc is defined by the intersection of the ellipse and the line from the center of the ellipse to the point (startingpointx , startingpointy) The end of the arc is similarly defined by the intersection of the ellipse and the line from the center of the ellipse to the point (endingpointx , endingpointy) For example: <pre>Ar c (0 , 0 , 100 , 100 , 95 , 0 , 5 , 0) ;</pre>	

Method Name	Description	See also
	 <p>The diagram shows a light brown circular arc. A dashed rectangle represents its bounding box. The top-left corner of the rectangle is labeled 'left, top' in red. The top-right corner is labeled 'startx, starty' in blue. The bottom-right corner is labeled 'right, bottom' in red. The bottom-left corner is labeled 'endx, endy' in blue. A dashed line connects the top-left and top-right corners, and another dashed line connects the top-right and bottom-right corners. A dashed line also connects the top-left and bottom-right corners, passing through the center of the arc.</p>	
arco (int left, int top, int right, int bottom, int startingpointx, int startingpointy, int endingpointx, int endingpointy)	<p>As for the arc method, except that a line is drawn from the current position to the starting point of the arc, and then the current position is updated to the end point of the arc</p>	
bezierto (int controlpoint1x, int controlpoint1y, int controlpoint2x, int controlpoint2y, int endpointx, int endpointy)	<p>Draws a bezier curve and updates the pen position</p>	
defSize(int width, int height)	<p>Sets the default size of the element</p> <p>This can appear in IF and ELSE clauses with different values in each, and causes the element to be resized automatically each time the values change; for example:</p> <pre> if (HasTag(" horizontal ", " true")) { def Si ze(100, 20); r ect angl e(0, 0, 100, 100); } el se { def Si ze(20, 100); r ect angl e(0, 0, 100, 100); } </pre> <p>The above example sets the shape to the specified default size each time the Tagged Value <i>horizontal</i> is changed</p>	

Method Name	Description	See also
	When this is set, (Alt+Z) also resizes the shape to the defined dimensions The minimum value for both int width and int height is 10	
drawnativeshape()	Causes Enterprise Architect to render the shape using its usual, non-Shapescript notation; subsequent drawing commands are super-imposed over the native notation This method is only enabled for element Shape Scripts; line Shape Scripts are not supported	
ellipse(int left, int top, int right, int bottom)	Draws an ellipse with extents defined by left , top , right and bottom	
endpath()	Ends the sequence of drawing commands that define a path	
fillandstrokepath()	Fills the previously defined path with the current fill color, then draws its outline with the current pen	
fillpath()	Fills the previously defined path with the current fill color	
hidelabel(string labelname)	Hides the label specified by labelname , where labelname is one of the following values: <ul style="list-style-type: none"> • middletoplabel • middlebottomlabel • lefttoplabel • leftbottomlabel • righttoplabel • rightbottomlabel 	
image(string imageld, int left, int top, int right, int bottom)	Draws the image that has the name imageld in the Image Manager The image must exist within the model in which the stereotype is used; if it does not already exist in the model, you must import it as reference data or select it from within a technology file If the image is in a technology file, it should have a filename of the format <i><technology ID>::<imagename>.<extension></i>	Reference data ^[239] Adding images in MDG technology ^[107]
lineto(int x, int y)	Draws a line from the current cursor position to a point specified by x and y , and then updates the pen cursor to that position	
moveto(int x, int y)	Moves the pen cursor to the point specified by x and y	
polygon(int centerx, int centery, int numberofsides,	Draws a regular polygon with center at the point (centerx , centery), and numberofsides number of sides	

Method Name	Description	See also
int radius, float rotation)		
print(string text)	Prints the specified text string You cannot change the font size, type or color of this text	
printifdefined(string propertyname, string truepart(, string falsepart))	Prints the <i>truepart</i> if the given property exists and has a non-empty value, otherwise prints the optional <i>falsepart</i> You cannot change the font size, type or color of this text	
println(string text)	Appends a line of text to the shape and a line break You cannot change the font size, type or color of this text	
printwrapped(string text)	Prints the specified text string, wrapped over multiple lines if the text is wider than its containing shape You cannot change the font size, type or color of this text	
rectangle(int left, int top, int right, int bottom)	Draws a rectangle with extents at left, top, right, bottom . Values are percentages	
roundrect(int left, int top, int right, int bottom, int abs_cornerwidth, int abs_cornerheight)	Draws a rectangle with rounded corners, with extents defined by left, top, right and bottom The size for the corners is defined by abs_cornerwidth and abs_cornerheight ; these values do not scale with the shape	
setdefaultcolors()	Returns the brush and pen color to the default settings, or to the user-defined colors if available	Color Queries ^[1099]
setfillcolor(int red, int green, int blue) setfillcolor(Color newColor)	Sets the fill color You can specify the required color by defining RGB values or using a color value returned by any of the Color Queries such as: Get User Fill Color ()	Color Queries ^[1099]
setfixedregion(int xStart, int yStart, int xEnd, int yEnd)	Fixes a region in a connector into which a subshape can be drawn, so that the subshape is not rescaled with the length or orientation of the connector line For an example, see the end of the table in the <i>Example Scripts</i> topic	Example Scripts ^[1104]
setlinestyle(string linestyle)	Changes the stroke pattern for commands that use the pen	

Method Name	Description	See also
	<p>string linestyle: has the following valid styles:</p> <ul style="list-style-type: none"> • solid • dash • dot • dashdot • dashdotdot 	
setorigin (string relativeTo , int xOffset , int yOffset)	<p>Positions floating text labels relative to the main shape</p> <p>relativeTo is one of N, NE, E, SE, S, SW, W, NW, CENTER</p> <p>xOffset and yOffset are in pixels, not percentage values, and can be negative</p>	
setpen (int red , int green , int blue (, int penwidth))	<p>Sets the pen to the defined color and optionally sets the pen width</p> <p>This method is only for line-drawing commands. It does not affect any text commands</p>	
setpencolor (int red , int green , int blue) setpencolor (Color newColor)	<p>Sets the pen color</p> <p>You can specify the required color by defining RGB values or using a color value returned by any of the Color Queries; for example:</p> <p style="padding-left: 40px;">Get User Fill Color ()</p> <p>This method is only for line-drawing commands. It does not affect any text commands</p>	Color Queries ^[1099]
setpenwidth (int penwidth)	<p>Sets the width of the pen. Pen width should be between 1 and 5</p> <p>This method is only for line-drawing commands. It does not affect any text commands</p>	
showlabel (string labelname)	<p>Reveals the hidden label specified by labelname, where labelname is one of the following values:</p> <ul style="list-style-type: none"> • middletoplabel • middlebottomlabel • lefttoplabel • leftbottomlabel • righttoplabel • rightbottomlabel 	
startcloudpath (puffWidth, puffHeight, noise)	<p>Similar to StartPath, except that it draws the path with cloud-like curved segments (<i>puffs</i>)</p> <p>Parameters:</p> <ul style="list-style-type: none"> • <i>float</i> puffWidth (default = 30), the horizontal distance between puffs • <i>float</i> puffHeight (default = 15), the vertical distance between puffs • <i>float</i> noise (default = 1.0), the randomization of the puffs' positions 	
startpath ()	<p>Starts the sequence of drawing commands that define a path</p>	

Method Name	Description	See also
<code>strokepath()</code>	Draws the outline of the previously defined path with the current pen	

6.10.1.3.3.4 Color Queries

Color queries can only be used to retrieve arguments for the *SetPenColor* and *SetFillColor* commands. These queries can be used in place of the arguments.

```
getUserFillColor()
getUserBorderColor()
getUserFontColor()
getUserPenSize()
```

```
shape main
{
    setfillcolor(getuserbordercolor());
    setpencolor(getuserfillcolor());

    rectangle(0, 0, 100, 100);
}
```

6.10.1.3.3.5 Conditional Branching

Shape Scripts provide condition branching with the *if else* statement, and query methods that evaluate to either *True* or *False*. See:

- [Syntax Grammar](#)^[1097] for IF statement syntax.
- [Query Methods](#)^[1099] for methods that can be used as the conditional expression for IF statements.
- [Example Scripts](#)^[1104] for an example.

6.10.1.3.3.6 Query Methods

Two query methods are available for seeing if the associated element has certain tags or properties; these methods can be used as the conditional expression for an *if else* statement.

Method	Description	See also
<code>boolean HasTag(string tagname, (string tagvalue))</code>	Returns true if the associated element has a tag value with the name <i>tagname</i> If the second parameter <i>tagvalue</i> is provided, the tag <i>tagname</i> must be present, and the value of the tag has to be equal to <i>tagvalue</i> for the method to return true	
<code>boolean HasProperty(string propertyname, (string propertyvalue))</code>	Returns true if the associated element has a property with the name <i>propertyname</i> If the second parameter <i>propertyvalue</i> is provided, the property must be present, and the value of the property has to be equal to <i>propertyvalue</i> for the method to return true	Display Element/Connector Properties ^[1099]

6.10.1.3.3.7 Display Element/Connector Properties

The commands `print`, `println`, and `printwrapped` all take a string parameter representing the text to be printed. Element and connector properties can be added to the text using the substitution macro `#propertyname#`.

For example: `println("name: #NAME#");`

In addition to the properties listed below, Tagged Values can also be displayed by prefixing the Tagged Value name with TAG:.

For example: `print("#TAG:condition#");`

You can also test against and display an element's *custom* properties in the same way as you do the named properties.

For example: `if(hasproperty("Name","Value"))`

...

and: `print("#Name#");`

Properties for Element Shape Scripts

- addin (value returned from an Add-In)
- alias
- author
- cardinality
- classifier
- classifier.alias
- classifier.metatype
- classifier.stereotype
- classifier.type
- complexity
- concurrency
- datecreated
- datemodified
- diagram.handdrawn
- diagram.mdgtype
- diagram.name
- diagram.stereotype
- diagram.type
- haslinkeddokument
- isabstract
- isactive
- iscomposite
- isembedded
- isinparent
- isleaf
- islocked
- isroot
- isspec
- istagged
- keywords
- language
- metatype
- multiplicity
- name
- notes
- packagename
- parentedge ("right", "left", "top", "bottom")
- parent.metatype
- persistence
- phase

- propertytype
- propertytype.alias
- propertytype.metatype
- propertytype.name
- propertytype.stereotype
- scope
- status
- stereotype
- type
- version
- visibility.

Properties for Connector Shape Scripts

- addin (value returned from an Add-In)
- alias
- diagram.connectornotation
- diagram.handdrawn
- diagram.mdgtype
- diagram.name
- diagram.stereotype
- diagram.type
- direction
- effect
- guard
- isroot
- isleaf
- name
- notes
- rotationdirection ("up", "down", "left", "right") (see [Example Scripts](#)⁽¹¹⁰²⁾)
- source.aggregation
- source.alias
- source.changeable
- source.constraints
- source.element.name
- source.element.stereotype
- source.metatype
- source.multiplicity
- source.multiplicityisordered
- source.name
- source.qualifiers
- source.stereotype
- source.targetscope
- stereotype
- subtype
- target.aggregation
- target.alias
- target.changeable
- target.constraints
- target.element.name
- target.element.stereotype
- target.metatype
- target.multiplicity
- target.multiplicityisordered

- target.name
- target.qualifiers
- target.stereotype
- target.targetscope
- type
- weight.

6.10.1.3.3.8 Sub-Shapes

Shapes can contain - and be composed of - other shapes.

Subshape Layout:

To set the layout type, the *layoutType* attribute must be set in the *initialization attributes* section of the script; in other words, before any of the methods are called. Valid values for this attribute are:

- **LeftRight** - Shapes with leftright layout position subshapes side by side, with the first added on the left, and subsequent subshapes to the right
- **TopDown** - TopDown places subshapes in a vertical arrangement, with the first shape added to the top and subsequent shapes added below
- **Border** - Border layout requires an additional argument to the *addsubshape* method to specify which region of the containing shape the subshape is to occupy: N, E, S, W or CENTER; each region can only be occupied by one subshape

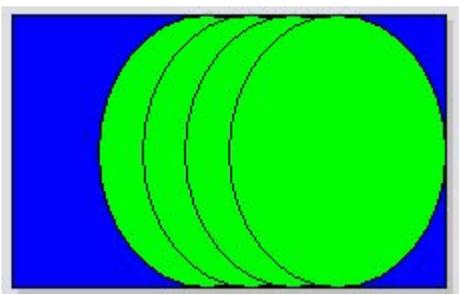
A subshape that is assigned to the E or W region must have its *preferredwidth* attribute specified in its declaration and, similarly, subshapes added to N or S must have their *preferredheight* attribute set; in this case, the values for these attributes are treated as static lengths and do not scale glyphs

For example:

```
shape main
{
  layouttype="topdown";
  setfillcolor(0,0,255);
  rectangle(0,0,100,100);
  addsubshape("sub",50,100,20,0);
  addsubshape("sub",50,100,30,-100);
  addsubshape("sub",50,100,40,-200);
  addsubshape("sub",50,100,50,-300);

  shape sub
  {
    setfillcolor(0,255,0);
    ellipse(0,0,100,100);
  }
}
```

The above script provides the following shape:



6.10.1.3.3.9 Reserved Names

Elements

Elements (such as Class, State or Event) have the following reserved names for parts of the shape.

Name	Description
shape main	The main shape is the whole shape.
shape label	The shape label gives the shape a detached label.
decoration <identifier>	Decoration gives the shape a decoration as defined by the name in <identifier>.

Connectors

Connectors (such as Association, Dependency or Generalization) have the following reserved names for parts of the shape.

Name	Description
shape main	The main shape is the whole shape.
shape source	The source shape is an extra shape at the source end of the connector.
shape target	The target shape is an extra shape at the target end of the connector.
shape <labelID>	The <labelID> gives the connector a detached label, where <labelID> is one of the following: <ul style="list-style-type: none"> • LeftTopLabel • MiddleTopLabel • RightTopLabel • LeftBottomLabel • MiddleBottomLabel • RightBottomLabel

6.10.1.3.3.10 Miscellaneous

Return Command

Execution of the script can be terminated by using the return command. Please see [Example Scripts](#) ¹¹⁰⁴ for an example.

Looping

The Shape Script feature does not support looping constructs.

Comments

C-style comments are supported. For example:

```
// C Style Single Line comment
/* Multi Line
comment supported */
```

String Manipulation

Not Supported.

Arithmetical Operations

Not Supported.

Variables Declaration

Not Supported.

Change ShapeScript Fonts

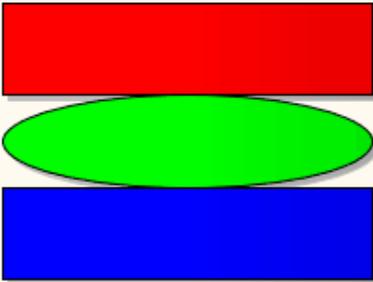
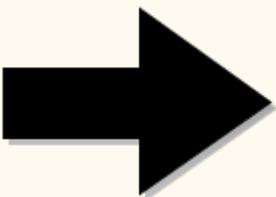
Not possible.

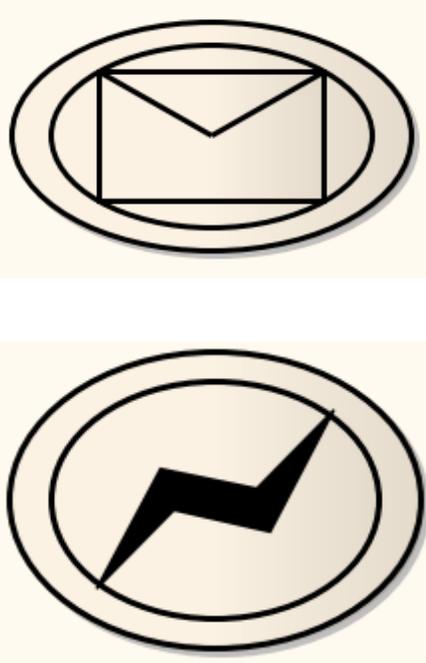
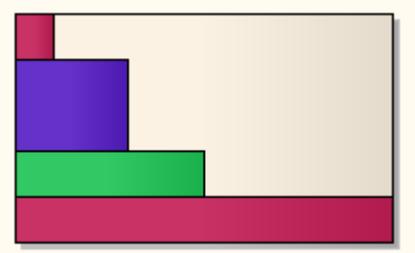
Can I apply a Shapescrpt without using Stereotypes?

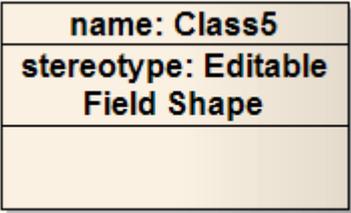
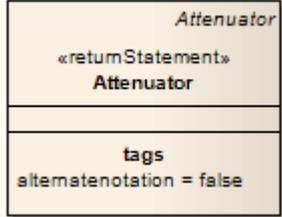
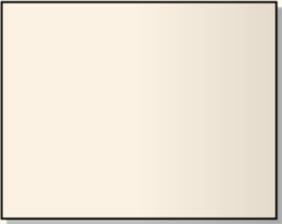
No.

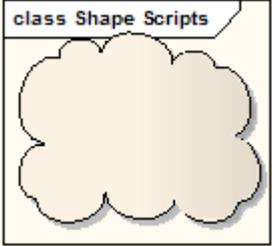
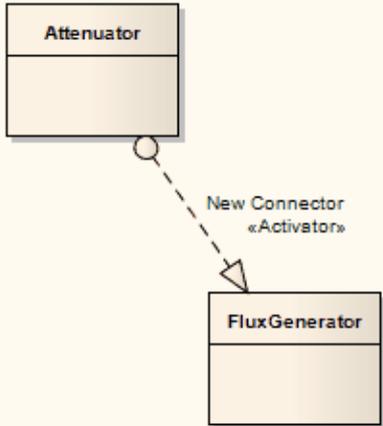
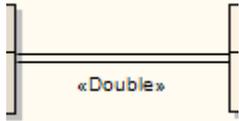
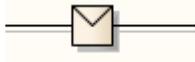
6.10.1.3.4 Example Scripts

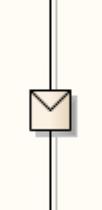
Below is a selection of example Shape Scripts.

Code	Result
<pre>// BASIC SHAPES shape main { setfillcolor(255,0,0); // (R,G,B) rectangle(0,0,90,30); // (x1,y1,x2,y2) setfillcolor(0,255,0); // (R,G,B) ellipse(0,30,90,60); // (x1,y1,x2,y2) setfillcolor(0,0,255); // (R,G,B) rectangle(0,60,90,90); // (x1,y1,x2,y2) }</pre>	
<pre>// SINGLE CONDITIONAL SHAPE shape main { if (HasTag("Trigger","Link")) { // Only draw if the object has a Tagged Value // Trigger=Link // Set the fill color for the path setfillcolor(0,0,0); startpath(); // Start to trace out a path moveto(23,40); } }</pre>	

Code	Result
<pre> lineto(23,60); lineto(50,60); lineto(50,76); lineto(76,50); lineto(50,23); lineto(50,40); endpath(); // End tracing out a path // Fill the traced path with the fill color fillandstrokepath(); return; } </pre>	
<pre> // MULTI CONDITIONAL SHAPE shape main { startpath(); ellipse(0,0,100,100); endpath(); fillandstrokepath(); ellipse(3,3,27,27); if (HasTag("Trigger", "None")) { return; } if (HasTag("Trigger", "Error")) { setfillcolor(0,0,0); startpath(); moveto(23,77); lineto(37,40); lineto(60,47); lineto(77,23); lineto(63,60); lineto(40,53); lineto(23,77); endpath(); fillandstrokepath(); return; } if (HasTag("Trigger", "Message")) { rectangle(22,22,78,78); moveto(22,22); lineto(50,50); lineto(78,22); return; } } </pre>	
<pre> // SUB SHAPES shape main { rectangle(0,0,100,100); addsubshape("red", 10, 20); addsubshape("blue", 30, 40); addsubshape("green", 50, 20); addsubshape("red", 100, 20); shape red { </pre>	

Code	Result
<pre> setfillcolor(200, 50, 100); rectangle(0,0,100,100); } shape blue { setfillcolor(100, 50, 200); rectangle(0,0,100,100); } shape green { setfillcolor(50, 200, 100); rectangle(0,0,100,100); } </pre>	
<pre> // EDITABLE FIELD SHAPE shape main { rectangle(0,0,100,100); addsubshape("namecompartment", 100, 20); addsubshape("stereotypercompartment", 100, 40); shape namecompartment { h_align = "center"; editablefield = "name"; rectangle(0,0,100,100); println("name: #name#"); } shape stereotypercompartment { h_align = "center"; editablefield = "stereotype"; rectangle(0,0,100,100); println("stereotype: #stereotype#"); } } </pre>	
<pre> // RETURN STATEMENT SHAPE shape main { if(hasTag("alternatenotation", "false")) { //draw ea's inbuild glyph drawnativeshape(); //exit script with the return statement return; } //alternate notation commands //... rectangle(0,0,100,100); } </pre>	 

Code	Result
<pre>// CLOUD PATH EXAMPLE SHAPE shape main { StartCloudPath(); Rectangle(0,0,100,100); EndPath(); FillAndStrokePath(); } </pre>	
<pre>// CONNECTOR SHAPE shape main { // draw a dashed line noshadow=true; setlinestyle("DASH"); moveto(0,0); lineto(100,0); } shape source { // draw a circle at the source end rotatable = true; startpath(); ellipse(0,6,12,-6); endpath(); fillandstrokepath(); } shape target { // draw an arrowhead at the target end rotatable = true; startpath(); moveto(0,0); lineto(16,6); lineto(16,-6); endpath(); fillandstrokepath(); } </pre>	
<pre>// DOUBLE LINE shape main { setlinestyle("DOUBLE"); moveto(0,0); lineto(100,0); } </pre>	
<pre>// ROTATION DIRECTION shape main { moveto(0,0); lineto(100,0); setfixedregion(40,-10,60,10); rectangle(40,-10,60,10); if(hasproperty("rotationdirection","up")) moveto(60,-10); lineto(50,0); lineto(60,10); } </pre>	

Code	Result
<pre> if(hasproperty("rotationdirection","down")) { moveto(40,-10); lineto(50,0); lineto(40,10); } if(hasproperty("rotationdirection","left")) { moveto(40,-10); lineto(50,0); lineto(60,-10); } if(hasproperty("rotationdirection","right")) { moveto(40,10); lineto(50,0); lineto(60,10); } } </pre>	

6.10.1.4 Tagged Value Types

Enterprise Architect provides a number of predefined Tagged Value Types that enable you to create your own:

- Tagged Values that are structured with a specific format, with or without tag filters, or
- Tagged Values that return values from the various reference data tables.

You can also use a masking parameter to create your own customized masked Tagged Value Type.

Notes:

- You can transport Tagged Value Type definitions between models, using the Export Reference Data and Import Reference Data options; Tagged Value Types are exported as *Property Types*

Learn More:

- [Predefined Tagged Value Types](#) ^[1108]
- [Predefined Tagged Value Type Filters](#) ^[1110]
- [Custom Tagged Value Type](#) ^[1113]
- [Reference Data](#) ^[1111]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

6.10.1.4.1 Predefined Structured Types

This table details the predefined structured Tagged Value types, along with the syntax used to create the initial values for their use. You use these to create your own structured Tagged Values.

Tagged Value Type and **Format** entries are case-sensitive.

Tagged Value Type	Format	Description	See also
AddinBroadcast	Type=AddinBroadcast: Values=YourAddinName;	Allows an AddIn to respond to an attempt to edit this Tagged Value by showing a dialog in which the value and notes can be edited	

Tagged Value Type	Format	Description	See also
Boolean	Type=Boolean; Default=Val;	Enables input of True or False , either of which can be the default value	
Classifier	Type=Classifier; Values=Type1,Type2; Stereotypes=Stereotype1;	Deprecated - use RefGUID and RefGUIDList Returns the <i>name</i> of a user-selected element from the model, where Type1 and Type2 specify one or more allowed element types and Stereotype1 represents an allowed stereotype	
Color	Type=Color; Default=Val;	Enables input of a color value from a color chooser menu, where the value is the decimal integer translation of the color's Hex RGB value For example, the RGB for Red is FF , and the decimal value is 255	
Const	Type=Const; Default=Val;	Enables creation of a read-only constant value	
Custom	Type= Custom;	Enables you to create your own template for predefined types; more information is provided in the Create Custom Tagged Value Type topic.	Create Custom Tagged Value Type <small>[1115]</small>
DateTime	Type=DateTime;	Enables input of the date and time for the Tagged Value from a calendar menu.	
Directory	Type=Directory; Default=Val;	Enables entry of a directory path from a browser. You can set a default directory path as a string value.	
Enum	Type=Enum; Values=Val1,Val2,Val3; Default=Val2;	Enables definition of a comma-separated list, where Val1 , Val2 and Val3 represent values in the list and Default represents the default value of the list.	
File	Type=File; Default=Val;	Enables input of a filename from a file browser dialog. The named file can be launched in its default application You can set a default file as a string containing the file path and file name	
Float, Decimal, Double	Type=Float; Type=Decimal; Type=Double; Default=Val;	Enable entry of a Float, Decimal or Double value. These types all map to the same type of data. You can set a default for any or all of these.	
Integer	Type=Integer; Default=Val;	Enables entry of an Integer value, and a default.	
Memo	Type=Memo;	Enables input of large and complex Tagged Values.	

Tagged Value Type	Format	Description	See also
RefGUID	Type=RefGUID; Values=Type1,Type2; Stereotypes=Stereotype1;	Enables the Tagged Value to reference an element from the model by specifying the element's <i>GUID</i> , where Type1 and Type2 specify one or more allowed diagram objects (such as Class , Component , Attribute or Operation) and Stereotype1 represents an allowed stereotype. Set the classifier, attribute or operation for a Tagged Value of this type by clicking on the (...) button against the Tagged Value in the Tagged Value window.	Classifier [690], Attribute or Operation [694]
RefGUIDList	Type=RefGUIDList; Values=Type1,Type2; Stereotypes=Stereotype1;	Enables the Tagged Value to reference a list of elements from the model by specifying each element's <i>GUID</i> , where Type1 and Type2 specify one or more allowed diagram objects (such as Class or Component) and Stereotype1 represents an allowed stereotype. Set the classifier, attribute or operation for a Tagged Value of this type by clicking on the (...) button against the Tagged Value in the Tagged Value window.	Classifier [690], Attribute or Operation [694]
Spin	Type=Spin; LowerBound=x; UpperBound=x; Default=Val;	Enables creation of a spin control with the value of LowerBound being the lowest value and UpperBound being the highest value. You can also set a default within that range.	
String	Type=String; Default=Val;	Enables entry of a string value, up to 255 characters in length, and a default text string. For longer texts, use Type=Memo.	Type= Memo [1109]
URL	Type=URL; Default=Val;	Enables entry of a web URL. The URL should start with: <ul style="list-style-type: none"> • 'http:/' • 'https:/' or • 'www.' You can set a default URL as a string value.	

Tag Filters:

The following table details filters that can be used to restrict where a Tagged Value can be applied.

Filter	Format	Description	See also
Applies To	AppliesTo=Type1,Type2;	Restricts the element types this filter can be applied to, where Type1 and Type2 are the valid types. Possible values are: <ul style="list-style-type: none"> • all element types 	

Filter	Format	Description	See also
		<ul style="list-style-type: none"> all connector types Attribute Operation and OperationParameter. 	
BaseStereotype	BaseStereotype=S1,S2;	Restricts the stereotypes that this tag belongs to, where S1 and S2 are the allowed stereotypes.	

Learn More:

- [Create Your Own Structured Tagged Values](#) ^[1111]

6.10.1.4.2 Create Structured Tagged Values

To create your own Tagged Value based on a predefined structured Tagged Value Type, follow the steps below:

Step	Description	See Also
1	Select the Settings UML Types menu option The UML Types dialog displays; select the Tagged Value Types tab	
2	Click on the New button	
3	In the Tag Name field type an appropriate Tagged Value name	
4	In the Description field type the purpose of the Tagged Value, if required	
5	In the Detail field copy-and-paste or type the syntax of the predefined structured Tagged Value Type	
6	Click on the Save button The Tagged Value type displays in the Defined Tag Types list	

Learn More:

- [Predefined Tagged Value Types](#) ^[1108]

6.10.1.4.3 Predefined Reference Data Types

This table details the predefined Reference Data Tagged Value types that are used to return the values held in a relevant table in Enterprise Architect, along with the syntax required for their use. You use these to create your own Reference Data Tagged Values.

Tagged Value Type	Format	Drop-Down List Returned	See also
Authors	Type=Enum; List=Authors;	Authors that have been defined for the Enterprise Architect model.	
Cardinality	Type=Enum; List=Cardinality;	Cardinality types that have been defined for the Enterprise Architect model.	
Clients	Type=Enum;	Clients that have been defined for the	

Tagged Value Type	Format	Drop-Down List Returned	See also
	List=Clients;	Enterprise Architect model.	
ComplexityTypes	Type=Enum; List=ComplexityTypes;	Complexity types that have been defined for the Enterprise Architect model. Whilst complexity types can be exported and imported as project reference data, they cannot be updated and so are effectively standard across all projects.	Reference data ^[77]
ConstraintTypes	Type=Enum; List=ConstraintTypes;	Constraint types that have been defined for the Enterprise Architect model.	
EffortTypes	Type=Enum; List=EffortTypes;	Effort types that have been defined for the Enterprise Architect model.	
MaintenanceTypes	Type=Enum; List=MaintenanceTypes;	Maintenance types that have been defined for the Enterprise Architect model.	
ObjectTypes	Type=Enum; List=ObjectTypes;	Object types that have been defined for the Enterprise Architect model.	
Phases	Type=Enum; List=Phases;	Phases that have been defined for the Enterprise Architect model.	
ProblemTypes	Type=Enum; List=ProblemTypes;	Problem types that have been defined for the Enterprise Architect model.	
RoleTypes	Type=Enum; List=RoleTypes;	Role types that have been defined for the Enterprise Architect model.	
RequirementTypes	Type=Enum; List=RequirementTypes;	Requirement types that have been defined for the Enterprise Architect model.	
Resources	Type=Enum; List=Resources;	Resources that have been defined for the Enterprise Architect model.	
RiskTypes	Type=Enum; List=RiskTypes;	Risk types that have been defined for the Enterprise Architect model.	
RTFTemplates	Type=Enum; List=RTFTemplates;	RTF Templates that have been defined for the Enterprise Architect model.	
ScenarioTypes	Type=Enum; List=ScenarioTypes;	Scenario types that have been defined for the Enterprise Architect model.	
TestTypes	Type=Enum; List=TestTypes;	Test types that have been defined for the Enterprise Architect model.	

Learn More

- [Create Your Own Reference Data Tagged Values](#)^[113]

6.10.1.4.4 Create Reference Data Tagged Values

To create your own Tagged Value based on a predefined Reference Data Tagged Value Type, follow the steps below:

Step	Description	See Also
1	Select the Settings UML Types menu option The UML Types dialog displays; select the Tagged Value Types tab	
2	In the Tag field type an appropriate Tagged Value name	
3	In the Description field type a description of the purpose of the Tagged Value, if required	
4	In the Detail field copy-and-paste or type the syntax of the predefined Reference Data Tagged Value Type This enables you to assign any of the previously defined Authors to a model feature	Model Elements and Features with Tagged Values ^[762]

Notes:

- If the values in the reference data are changed after the Tagged Value Type is created, Enterprise Architect must be reloaded in order to reflect the changes in the Tagged Value Type

Learn More:

- [Predefined Reference Data](#) ^[1117]

6.10.1.4.5 Create Custom Tagged Value Type

Custom masked Tagged Values give you great flexibility in designing model components that accept data entries.

How to:

To create a masked Tagged Value follow the steps below:

Step	Action	See Also
1	Select the Settings UML Types menu option The UML Types dialog displays; select the Tagged Value Types tab	
2	In the Tag field type an appropriate name for the Tagged Value	
3	In the Description field type the purpose of the Tagged Value, if required	
4	In the Detail field type Type=Custom;	

The type **Custom** enables you to set up the appropriate mask, using the following characters to define the format of the mask:

Mask	Description
D	Enables the Tagged Value to display digits only.
d	Enables the Tagged Value to display digits or spaces.
+	Enables the use of +, - or spaces.
C	Enables the use of alpha characters only.
c	Enables the Tagged Value to be an alpha character or a space.
A	Enables the use of alphanumeric characters.
a	Enables the Tagged Value to use alphanumeric values or a space.

In the diagram below the **Mask** configuration option shows syntax that first defines seven blank spaces, which are occupied by characters determined by the template option. The first two visible characters in the **Mask** option are represented by a lower case **c** indicating that the enableable information can entered as either an alpha character or as a space. The following blank spaces again indicate space defined by the template option and the remaining characters are defined by the **d** character, which represents the enableable characters as digits or spaces. The hyphen is present in the final output, splitting up the digits.

With the **Template** configuration option, the syntax defines the template of the masked option by occupying the blank spaces that are present in the **Mask** option. The template is used to ensure that this information is present with every use of this custom Tagged Value. The underscored values indicate the area that is to be occupied by data input by the user as defined in the **Mask** option.

The screenshot shows the 'Tagged Value Types' configuration window. The 'Tag Name' is 'MemberZip' and the 'Description' is 'Zip Code'. The 'Detail' section contains the following configuration:

```
Type=Custom:
Mask= cc dddd.dddd;
Template=State: __Zip: ____-____;
```

Below the detail section are 'New', 'Save', and 'Delete' buttons. At the bottom, a table lists 'Defined Tag Types' with columns 'Type' and 'Description'. The 'MemberZip' entry is highlighted in blue.

Type	Description
IORules	IORules
MemberZip	Zip Code
Message	
OutMessage	

6.10.1.5 Code Template Framework

The Code Template Framework (CTF) is used during forward engineering of UML models. This section discusses how you customize the way in which Enterprise Architect generates source code, using the [Code Template Editor](#).^[114]

Enterprise Architect's code templates specify the transformation from UML elements to the various parts of a given programming language. The templates are written as plain text with a [syntax](#).^[115] that shares some

aspects of both mark-up languages and scripting languages. The Base Templates provided in Enterprise Architect are described in the [Base Templates](#) topic.

6.10.1.5.1 Code Template Syntax

Code Templates are written as plain text, using Enterprise Architect's code template editor (see [The Code Template Editor](#)). The template syntax centers on three basic constructs:

- [Literal Text](#)
- [Macros](#)
- [Variables](#)

Templates can contain any or all of these constructs.

6.10.1.5.1.1 Literal Text

All text within a given template that is not part of a macro or a variable definition/reference, is considered literal text. With the exception of blank lines, which are ignored, literal text is directly substituted from the template into the generated code.

Consider the following excerpt from the java Class Declaration template:

```
%PI =" " %
%CONVERT_SCOPE( classScope) %

%classStereotype=="static" ? "static": "" %
%classStereotype=="final" ? "final": "" %
%classStereotype=="static final" ? "static final": "" %
%classAbstract=="T" ? "abstract": "" %
%PI =" " %
class %className%$bases
```

On the final line, the word *class*, including the subsequent space, would be treated as literal text and thus reproduced in the output. The blank line following the *CONVERT_SCOPE* macro, however, would have no effect on the output.

The %, \$ and " characters have special meaning in the template syntax and cannot always be used as literal text. If these characters must be generated from within the templates, they can be safely reproduced using the following direct substitution macros:

Macro	Action
%dl%	Produce a literal \$ character.
%pc%	Produce a literal % character.
%qt%	Produce a literal " character.

6.10.1.5.1.2 Macros

Macros provide access to element fields within the UML model and are also used to structure the generated output. All macros are enclosed within percent (%) signs. The CTF contains six basic types of macros:

- [Template substitution macros](#)
- [Field substitution macros](#)
- [Tagged Value substitution macros](#)
- [Control macros](#)
- [Function macros](#)
- [EASL code generation macros](#)

In general, macros (including the % delimiters) are substituted with literal text in the output. For example

consider the following item from the *Class Declaration* template:

```
... class %className% ...
```

The field substitution macro, `%className%`, would result in the current Class name being substituted in the output. So if the Class being generated was named *Foo*, the output would be:

```
... class Foo ...
```

Template substitution macros correspond to [Base templates](#)^[1483]. These macros result in the execution of the named template. By convention, template macros are named according to Pascal casing.

Structure: %<TemplateName>%

where `<TemplateName>` can be one of the templates listed below.

When a template is referenced from within another template, it is generated with respect to the elements currently in scope. The specific template is selected based on the stereotypes of the elements in scope.

As noted previously, there is an implicit hierarchy among the various templates. Some care should be taken in order to preserve a sensible hierarchy of template references. For example, it does not make sense to use the `%ClassInherits%` macro within any of the attribute or operation templates. Conversely, the *Operation* and *Attribute* templates are designed for use within the *ClassBody* template.

The CTF contains the following template substitution macros:

- AttributeDeclaration
- AttributeNotes
- Attribute
- Class
- ClassImpl
- ClassBase
- ClassBody
- ClassBodyImpl
- ClassDeclaration
- ClassDeclarationImpl
- ClassInherits
- ClassInterface
- ClassNotes
- ClassParameter
- File
- FileImpl
- ImportSection
- ImportSectionImpl
- InnerClass
- InnerClassImpl
- LinkedAttribute
- LinkedAttributeNotes
- LinkedAttributeDeclaration
- LinkedClassBase
- LinkedClassInterface
- Namespace
- NamespaceBody
- NamespaceDeclaration
- NamespaceImpl
- Operation
- OperationBody
- OperationBodyImpl
- OperationDeclaration
- OperationDeclarationImpl
- OperationImpl
- OperationNotes
- Parameter

The *field substitution* macros provide access to data in the model. In particular, they are used to access data fields from:

- Packages
- Classes
- Attributes
- Operations
- Parameters.

Field substitution macros are named according to Camel casing. By convention, the macro is prefixed with an abbreviated form of the corresponding model element. For example, attribute-related macros begin with **att**, as in the `%attName%` macro, to access the name of the attribute in scope.

Macros that represent checkboxes return a value of T if the box is selected. Otherwise the value is empty.

The following table lists each of the field substitution macros with a description of the result.

Macro Name	Description
attAlias	Attributes dialog: Alias .
attAllowDuplicates	Attributes Detail dialog: Allow Duplicates checkbox.
attClassifierGUID	The unique GUID for the classifier of the current attribute.
attCollection	Attributes Detail dialog: Attribute is a Collection checkbox.
attConst	Attributes dialog: Const checkbox.
attContainerType	Attributes Detail dialog: Container Type .
attContainment	Attributes dialog: Containment .
attDerived	Attributes dialog: Derived checkbox.
attGUID	The unique GUID for the current attribute.
attInitial	Attributes dialog: Initial .
attIsEnumLiteral	Attributes dialog: Is Literal checkbox.
attLength	Column dialog: Length .
attLowerBound	Attributes Detail dialog: Lower Bound .
attName	Attributes dialog: Name .
attNotes	Attributes dialog: Notes .
attOrderedMultiplicity	Attributes Detail dialog: Ordered Multiplicity checkbox.
attProperty	Attributes dialog: Property checkbox.
attQualType	The attribute type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the attribute classifier has not been set, is equivalent to the <i>attType</i> macro.
attScope	Attributes dialog: Scope .
attStatic	Attributes dialog: Static checkbox.
attStereotype	Attributes dialog: Stereotype .
attType	Attributes dialog: Type .
attUpperBound	Attributes Detail dialog: Upper Bound .
attVolatile	Attributes Detail dialog: Transient checkbox.
classAbstract	Class dialog: Abstract checkbox.
classAlias	Class dialog: Alias .
classArguments	Class Detail dialog: C++ Templates: Arguments .
classAuthor	Class dialog: Author .
classBaseName	Type Hierarchy dialog: Class Name (for use where no connector exists between child and base Classes).

Macro Name	Description
classBaseScope	The scope of the inheritance as reverse engineered. (For use where no connector exists between child and base Classes.)
classBaseVirtual	The virtual property of the inheritance as reverse engineered. (For use where no connector exists between child and base Classes.)
classComplexity	Class dialog: Complexity .
classCreated	The date and time the Class was created.
classGUID	The unique GUID for the current Class.
classHasConstructor	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns T if one is a default constructor. Typically used with the genOptGenConstructor _[1123] macro.
classHasCopyConstructor	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns T if one is a copy constructor. Typically used with the genOptGenCopyConstructor _[1123] macro.
classHasDestructor	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns T if one is a destructor. Typically used with the genOptGenDestructor _[1123] macro.
classHasParent	True , if the Class in scope has one or more base Classes.
classImports	Code Gen dialog: Imports .
classIsActive	Class Advanced dialog: Is Active checkbox.
classIsInstantiated	True , if the Class is an instantiated template Class.
classIsLeaf	Class Advanced dialog: Is Leaf checkbox.
classIsRoot	Class Advanced dialog: Is Root checkbox.
classIsSpecification	Class Advanced dialog: Is Specification checkbox.
classKeywords	Class dialog: Keywords .
classLanguage	Class dialog: Language .
classMacros	A space separated list of macros defined for the Class.
classModified	The date and time the Class was last modified.
classMultiplicity	Class Advanced dialog: Multiplicity .
className	Class dialog: Name .
classNotes	Class dialog: Note .
classParamDefault	Class Detail dialog.
classParamName	Class Detail dialog.
classParamType	Class Detail dialog.
classPersistence	Class dialog: Persistence .
classPhase	Class dialog: Phase .

Macro Name	Description
classQualName	The Class name prefixed by its outer Classes. Class names are separated by double colons (::).
classScope	Class dialog: Scope .
classStereotype	Class dialog: Stereotype .
classStatus	Class dialog: Status .
classVersion	Class dialog: Version .
connectorAlias	Connector Properties dialog: Alias .
connectorDestAccess	Connector Properties dialog, Target Role tab: Access .
connectorDestAggregation	Connector Properties dialog, Target Role tab: Aggregation .
connectorDestAlias	Connector Properties dialog, Target Role tab: Alias .
connectorDestAllowDuplicates	Connector Properties dialog, Target Role tab: Allow Duplicates checkbox.
connectorDestChangeable	Connector Properties dialog, Target Role tab: Changeable .
connectorDestConstraint	Connector Properties dialog, Target Role tab: Constraint(s) .
connectorDestContainment	Connector Properties dialog, Target Role tab: Containment .
connectorDestDerived	Connector Properties dialog, Target Role tab: Derived checkbox.
connectorDestDerivedUnion	Connector Properties dialog, Target Role tab: DerivedUnion checkbox.
connectorDestElem*	A set of macros that access a property of the element at the target end of a connector. The * (asterisk) is a wildcard that corresponds to any class substitution macro in this list; for example: <i>connectorDestElemAlias (classAlias)</i> , <i>connectorDestElemAuthor (classAuthor)</i> .
connectorDestElemType	The element type of the connector destination element. (Separate from the <i>connectorDestElem*</i> macros because there is no <i>classType</i> substitution macro.)
connectorDestMemberType	Connector Properties dialog, Target Role tab: Member Type .
connectorDestMultiplicity	Connector Properties dialog, Target Role tab: Multiplicity .
connectorDestNavigability	Connector Properties dialog, Target Role tab: Navigability .
connectorDestNotes	Connector Properties dialog, Target Role tab: Role Notes .
connectorDestOrdered	Connector Properties dialog, Target Role tab: Ordered checkbox.
connectorDestOwned	Connector Properties dialog, Target Role tab: Owned checkbox.
connectorDestQualifier	Connector Properties dialog, Target Role tab: Qualifier(s) .
connectorDestRole	Connector Properties dialog, Target Role tab: Role .
connectorDestScope	Connector Properties dialog, Target Role tab: Target Scope .
connectorDestStereotype	Connector Properties dialog, Target Role tab: Stereotype .

Macro Name	Description
connectorDirection	Connector Properties : Direction .
connectorEffect	Transition Constraints dialog: Effect .
connectorGuard	Object Flow and Transition Constraints dialog: Guard .
connectorGUID	The unique GUID for the current connector.
connectorName	Connector Properties : Name .
connectorNotes	Connector Properties : Notes .
connectorSourceAccess	Connector Properties dialog, Source Role tab: Access .
connectorSourceAggregation	Connector Properties dialog, Source Role tab: Aggregation .
connectorSourceAlias	Connector Properties dialog, Source Role tab: Alias .
connectorSourceAllowDuplicates	Connector Properties dialog, Source Role tab: Allow Duplicates checkbox.
connectorSourceChangeable	Connector Properties dialog, Source Role tab: Changeable .
connectorSourceConstraint	Connector Properties dialog, Source Role tab: Constraint(s) .
connectorSourceContainment	Connector Properties dialog, Source Role tab: Containment .
connectorSourceDerived	Connector Properties dialog, Source Role tab: Derived checkbox.
connectorSourceDerivedUnion	Connector Properties dialog, Source Role tab: DerivedUnion checkbox.
connectorSourceElem*	A set of macros that access a property of the element at the source end of a connector. The * (asterisk) is a wildcard that corresponds to any class substitution macro in this list; for example: <i>connectorSourceElemAlias (classAlias)</i> , <i>connectorSourceElemAuthor (classAuthor)</i> .
connectorSourceElemType	The element type of the connector source element. (Separate from the <i>connectorSourceElem*</i> macros because there is no <i>classType</i> substitution macro.)
connectorSourceMemberType	Connector Properties dialog, Source Role tab: Member Type .
connectorSourceMultiplicity	Connector Properties dialog, Source Role tab: Multiplicity .
connectorSourceNavigability	Connector Properties dialog, Source Role tab: Navigability .
connectorSourceNotes	Connector Properties dialog, Source Role tab: Role Notes .
connectorSourceOrdered	Connector Properties dialog, Source Role tab: Ordered checkbox.
connectorSourceOwned	Connector Properties dialog, Source Role tab: Owned checkbox.
connectorSourceQualifier	Connector Properties dialog, Source Role tab: Qualifier(s) .
connectorSourceRole	Connector Properties dialog, Source Role tab: Role .
connectorSourceScope	Connector Properties dialog, Source Role tab: Target Scope .
connectorSourceStereotype	Connector Properties dialog, Source Role tab: Stereotype .
connectorStereotype	Connector Properties dialog: Stereotype .

Macro Name	Description
connectorTrigger	Transition Constraints dialog: Trigger .
connectorType	The connector type; for example, Association or Generalization.
connectorWeight	Object Flow Constraints dialog: Weight .
constraintName	Class dialog, Constraints tab: Name .
constraintNotes	Class dialog, Constraints tab: Notes .
constraintStatus	Class dialog, Constraints tab: Status .
constraintType	Class dialog, Constraints tab: Type .
constraintWeight	Class dialog, Constraints tab: ordering (hand up/down) keys.
eaDateTime	The current time with format: <i>DD-MMM-YYYY HH:MM:SS AM/PM</i> .
eaGUID	A unique GUID for this generation.
eaVersion	Program Version (Located in an Enterprise Architect dialog by selecting Help About EA).
effortName	Project Management window: Effort .
effortNotes	Project Management window: Notes (unlabelled).
effortTime	Project Management window: Time .
effortType	Project Management window: Type .
elemType	The element type: Interface or Class.
fileExtension	The file type extension of the file being generated.
fileName	The name of the file being generated.
fileNameImpl	The filename of the implementation file for this generation, if applicable.
fileHeaders	Code Gen dialog: Headers .
fileImports	Code Gen dialog: Imports . For supported languages this also includes dependencies derived from associations.
filePath	The full path of the file being generated.
filePathImpl	The full path of the implementation file for this generation, if applicable.
genOptActionScriptVersion	ActionScript Specifications dialog: Default Version .
genOptCDefaultAttributeType	C Specifications dialog: Default Attribute Type .
genOptCGenMethodNotesInBody	C Specifications dialog: Method Notes In Implementation .
genOptCGenMethodNotesInHeader	C Specifications dialog: Method Notes In Header .
genOptCSynchNotes	C Specifications dialog: Synchronize Notes in Generation .
genOptCSynchCFile	C Specifications dialog: Synchronise Implementation file in Generation .

Macro Name	Description
genOptCDefaultSourceDirectory	C Specifications dialog: Default Source Directory .
genOptCNamespaceDelimiter	C Specifications dialog: Namespace Delimiter .
genOptCOperationRefParam	C Specifications dialog: Reference as Operation Parameter .
genOptCOperationRefParamStyle	C Specifications dialog: Reference Parameter Style .
genOptCOperationRefParamName	C Specifications dialog: Reference Parameter Name .
genOptCConstructorName	C Specifications dialog: Default Constructor Name .
genOptCDestructorName	C Specifications dialog: Default Destructor Name .
genOptCPPCommentStyle	C++ Specifications dialog: Comment Style .
genOptCPPDefaultAttributeType	C++ Specifications dialog: Default Attribute Type .
genOptCPPDefaultReferenceType	C++ Specifications dialog: Default Reference Type .
genOptCPPDefaultSourceDirectory	C++ Specifications dialog: Default Source Directory .
genOptCPPGenMethodNotesInHeader	C++ Specifications dialog: Method Notes In Header checkbox.
genOptCPPGenMethodNotesInBody	C++ Specifications dialog: Method Notes In Body checkbox.
genOptCPPGetPrefix	C++ Specifications dialog: Get Prefix .
genOptCPPHeaderExtension	C++ Specifications dialog: Header Extension .
genOptCPPSetPrefix	C++ Specifications dialog: Set Prefix .
genOptCPPSourceExtension	C++ Specifications dialog: Source Extension .
genOptCPPSynchCPPFile	C++ Specifications dialog: Synchronize Notes .
genOptCPPSynchNotes	C++ Specifications dialog: Synchronize CPP File .
genOptCSDefaultAttributeType	C# Specifications dialog: Default Attribute Type .
genOptCSSourceExtension	C# Specifications dialog: Default file extension .
genOptCSGenDispose	C# Specifications dialog: Generate Dispose .
genOptCSGenFinalizer	C# Specifications dialog: Generate Finalizer .
genOptCSGenNamespace	C# Specifications dialog: Generate Namespace .
genOptCSDefaultSourceDirectory	C# Specifications dialog: Default Source Directory .
genOptDefaultAssocAttName	Attribute Specifications dialog: Default name for associated attrib.
genOptDefaultConstructorScope	Object Lifetimes dialog: Default Constructor Visibility .
genOptDefaultCopyConstructorScope	Object Lifetimes dialog: Default Copy Constructor Visibility .
genOptDefaultDatabase	Code Editors dialog: Default Database .

Macro Name	Description
genOptDefaultDestructorScope	Object Lifetimes dialog: Default Destructor Constructor Visibility .
genOptGenCapitalisedProperties	Source Code Engineering dialog: Capitalize Attribute Names for Properties checkbox.
genOptGenComments	Source Code Engineering dialog: Generate Comments checkbox.
genOptGenConstructor	Object Lifetimes dialog: Generate Constructor checkbox.
genOptGenConstructorInline	Object Lifetimes dialog: Constructor Inline checkbox.
genOptGenCopyConstructor	Object Lifetimes dialog: Generate Copy Constructor checkbox.
genOptGenCopyConstructorInline	Object Lifetimes dialog: Copy Constructor Inline checkbox.
genOptGenDestructor	Object Lifetimes dialog: Generate Destructor checkbox.
genOptGenDestructorInline	Object Lifetimes dialog: Destructor Inline checkbox.
genOptGenDestructorVirtual	Object Lifetimes dialog: Virtual Destructor checkbox.
genOptGenImplementedInterfaceOps	Attribute/Operations Specifications dialog: Generate methods for implemented interfaces checkbox.
genOptGenPrefixBoolProperties	Source Code Engineering dialog: Use is prefix for boolean property Get() .
genOptGenRoleNames	Source Code Engineering dialog: Autogenerate role names when creating code .
genOptGenUnspecAssocDir	Source Code Engineering dialog: Do not generate members where Association direction is unspecified checkbox.
genOptJavaDefaultAttributeType	Java Specifications dialog: Default attribute type .
genOptJavaGetPrefix	Java Specifications dialog: Get Prefix .
genOptJavaDefaultSourceDirectory	Java Specifications dialog: Default Source Directory .
genOptJavaSetPrefix	Java Specifications dialog: Set Prefix .
genOptJavaSourceExtension	Java Specifications dialog: Source code extension .
genOptPHPDefaultSourceDirectory	PHP Specifications dialog: Default Source Directory .
genOptPHPGetPrefix	PHP Specifications dialog: Get Prefix .
genOptPHPSetPrefix	PHP Specifications dialog: Set Prefix .
genOptPHPSourceExtension	PHP Specifications dialog: Default file extension .
genOptPHPVersion	PHP Specifications dialog: PHP Version .
genOptPropertyPrefix	Source Code Engineering dialog: Remove prefixes when generating Get/Set properties .
genOptVBMultiUse	VB Specifications dialog: Multiuse checkbox.
genOptVBPersistable	VB Specifications dialog: Persistable checkbox.
genOptVBDataBindingBehavior	VB Specifications dialog: Data binding behavior checkbox.

Macro Name	Description
genOptVBDataSourceBehavior	VB Specifications dialog: Data source behavior checkbox.
genOptVBGlobal	VB Specifications dialog: Global namespace checkbox.
genOptVBCreatable	VB Specifications dialog: Creatable checkbox.
genOptVBExposed	VB Specifications dialog: Exposed checkbox.
genOptVBMTS	VB Specifications dialog: MTS Transaction Mode .
genOptVBNetGenNamespace	VB.Net Specifications dialog: Generate Namespace .
genOptVBVersion	VB Specifications dialog: Default Version .
genOptWrapComment	Source Code Engineering dialog: Wrap length for comment lines .
importClassName	The name of the Class being imported.
importFileName	The filename of the Class being imported.
importFilePath	The full path of the Class being imported.
importFromAggregation	T if the Class has an Aggregation connector to a Class in this file, F otherwise.
importFromAssociation	T if the Class has an Association connector to a Class in this file, F otherwise.
importFromAtt	T if an attribute of a Class in the current file is of the type of this Class, F otherwise.
importFromDependency	T if the Class has a Dependency connector to a Class in this file, F otherwise.
importFromGeneralization	T if the Class has a Generalization connector to a Class in this file, F otherwise.
importFromMeth	T if a method return type of a Class in the current file is the type of this Class, F otherwise.
importFromParam	T if a method parameter of a Class in the current file is of the type of this Class, F otherwise.
importFromRealization	T if the Class has a Realization connector to a Class in this file, F otherwise.
importInFile	T if the Class is in the current file, F otherwise.
importPackagePath	The package path with a '.' separator of the Class being imported.
ImportRelativeFilePath	The relative file path of the Class being imported from the file path of the file being generated.
linkAttAccess	Association Properties Target Role dialog: Access .
linkAttCollectionClass	The collection appropriate for the linked attribute in scope.
linkAttContainment	Association Properties Target Role dialog: Containment .
linkAttName	Association Properties dialog: Target .
linkAttNotes	Association Properties Target Role dialog: Role Notes .

Macro Name	Description
linkAttQualName	The Association target qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited).
linkAttRole	Association Properties Target Role dialog: Role .
linkAttStereotype	Association Properties Target Role dialog: Stereotype .
linkAttTargetScope	Association Properties Target Role dialog: Target Scope .
linkCard	Link Properties Target Role dialog: Multiplicity .
linkedFileLastWrite	Class Properties dialog: Last Write .
linkedFileNotes	Class Properties dialog: Notes .
linkedFilePath	Class Properties dialog: File Path .
linkedFileSize	Class Properties dialog: Size .
linkedFileType	Class Properties dialog: Type .
linkGUID	The unique GUID for the current connector.
linkParentName	Generalization Properties dialog: Target .
linkParentQualName	The Generalization target qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited).
linkStereotype	The stereotype of the current connector.
linkVirtualInheritance	Generalization Properties dialog: Virtual Inheritance .
metricName	Project Management dialog, Metrics tab: Metric field.
metricNotes	Project Management dialog, Metrics tab: (Notes) field.
metricType	Project Management dialog, Metrics tab: Type field.
metricWeight	Project Management dialog, Metrics tab: Weight field.
opAbstract	Operation dialog: Virtual checkbox.
opAlias	Operation dialog: Alias .
opBehavior	Operation Behavior dialog: Behavior .
opCode	Operation Behavior dialog: Initial Code .
opConcurrency	Operation dialog: Concurrency .
opConst	Operation dialog: Const checkbox.
opGUID	The unique GUID for the current operation.
opImplMacros	A space-separated list of macros defined in the implementation of this operation.
opIsQuery	Operation dialog: IsQuery checkbox.
opMacros	A space-separated list of macros defined in the declaration for this operation.
opName	Operation dialog: Name .

Macro Name	Description
opNotes	Operation dialog: Notes .
opPure	Operation dialog: Pure checkbox.
opReturnArray	Operation dialog: Return Array checkbox.
opReturnClassifierGUID	The unique GUID for the classifier of the current operation.
opReturnQualType	The operation return type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the return type classifier has not been set, is equivalent to the <i>opReturnType</i> macro.
opReturnType	Operation dialog: Return Type .
opScope	Operation dialog: Scope .
opStatic	Operation dialog: Static checkbox.
opStereotype	Operation dialog: Stereotype .
opSynchronized	Operation dialog: Synchronized checkbox.
packageAbstract	Package dialog: Abstract .
packageAlias	Package dialog: Alias .
packageAuthor	Package dialog: Author .
packageComplexity	Package dialog: Complexity .
packageGUID	The unique GUID for the current package.
packageKeywords	Package dialog: Keywords .
packageLanguage	Package dialog: Language .
packageName	Package dialog: Name .
packagePath	The string representing the hierarchy of packages, for the Class in scope. Each package name is separated by a dot (.).
packagePhase	Package dialog: Phase .
packageScope	Package dialog: Scope .
packageStatus	Package dialog: Status .
packageStereotype	Package dialog: Stereotype .
packageVersion	Package dialog: Version .
paramClassifierGUID	The unique GUID for the classifier of the current parameter.
paramDefault	Operation Parameters dialog: Default .
paramFixed	Operation Parameters dialog: Fixed checkbox.
paramGUID	The unique GUID for the current parameter.
paramIsEnum	True , if the parameter uses the <i>enum</i> keyword (C++).
paramKind	Operation Parameters dialog: Kind .

Macro Name	Description
paramName	Operation Parameters dialog: Name .
paramNotes	Operation Parameters dialog: Notes .
paramQualType	The parameter type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the parameter classifier has not been set, is equivalent to the <i>paramType</i> macro.
paramType	Operation Parameters dialog: Type .
problemCompletedBy	Maintenance dialog, Element Issues tab: Completed by .
problemCompletedDate	Maintenance dialog, Element Issues tab: Completed .
problemHistory	Maintenance dialog, Element Issues tab: History .
problemName	Maintenance dialog, Element Issues tab: Name .
problemNotes	Maintenance dialog, Element Issues tab: Description .
problemPriority	Maintenance dialog, Element Issues tab: Priority .
problemRaisedBy	Maintenance dialog, Element Issues tab: Raised by .
problemRaisedDate	Maintenance dialog, Element Issues tab: Raised .
problemStatus	Maintenance dialog, Element Issues tab: Status .
problemVersion	Maintenance dialog, Element Issues tab: Version .
requirementDifficulty	Properties dialog: Require tab: Difficulty .
requirementLastUpdated	Properties dialog: Require tab: Last Update .
requirementName	Properties dialog: Require tab: Short Description .
requirementNotes	Properties dialog: Require tab: Notes .
requirementPriority	Properties dialog: Require tab: Priority .
requirementStatus	Properties dialog: Require tab: Status .
requirementType	Properties dialog: Require tab: Type .
resourceAllocatedTime	Project Management window, Resource Allocation tab: Allocated Time .
resourceEndDate	Project Management window, Resource Allocation tab: End Date .
resourceExpectedTime	Project Management window, Resource Allocation tab: Expected Time .
resourceExpendedTime	Project Management window, Resource Allocation tab: Time Expended .
resourceHistory	Project Management window, Resource Allocation tab: History .
resourceName	Project Management window, Resource Allocation tab: Resource .
resourceNotes	Project Management window, Resource Allocation tab: Description .
resourcePercentCompleted	Project Management window, Resource Allocation tab: Completed (%) .

Macro Name	Description
resourceRole	Project Management window, Resource Allocation tab: Role .
resourceStartDate	Project Management window, Resource Allocation tab: Start Date .
riskName	Project Management window, Risks tab: Risk .
riskNotes	Project Management window, Risks tab: (Notes).
riskType	Project Management window, Risks tab: Type .
riskWeight	Project Management window, Risks tab: Weight .
scenarioGUID	The unique ID for a scenario. Identifies the scenario unambiguously within a model.
scenarioName	Properties dialog, Scenario tab: Scenario .
scenarioNotes	Properties dialog, Scenario tab: (Notes).
scenarioType	Properties dialog, Scenario tab: Type .
testAcceptanceCriteria	Testing window: Acceptance Criteria .
testCheckedBy	Testing window: Checked By .
testDateRun	Testing window: Last Run .
testClass	The Testing window tab (the type of test defined): Unit, Integration, System, Acceptance, Scenario .
testInput	Testing window: Input .
testName	Testing window: Test .
testNotes	Testing window: Description .
testResults	Testing window: Results .
testRunBy	Testing window: Run By .
testStatus	Testing window: Status .
testType	Testing window: Type .

Field substitution macros can be used in one of two ways:

Use 1: Direct Substitution

This form directly substitutes the corresponding value of the element in scope into the output.

Structure: %<macroName>%

Where <macroName> can be any of the macros listed above.

Examples:

- %className%
- %opName%
- %attName%

Use 2: Conditional Substitution

This form of the macro enables alternative substitutions to be made depending on the macro's value.

Structure: `%<macroName> (== " <test> ") ? <subTrue> (: <subFalse>) %`

Where:

- `(<text>)` denotes that `<text>` is optional
- `<test>` is a string representing a possible value for the macro
- `<subTrue>` and `<subFalse>` can be a combination of quoted strings and the keyword value; where the value is used, it is replaced with the macro's value in the output.

Examples:

- `%classAbstract=="T" ? "pure" : ""%`
- `%opStereotype=="operator" ? "operator" : ""%`
- `%paramDefault != "" ? " = " value : ""%`

The above three examples output nothing if the condition fails. In this case the false condition can be omitted, resulting in the following usage:.

Examples:

- `%classAbstract=="T" ? "pure" %`
- `%opStereotype=="operator" ? "operator" %`
- `%paramDefault ? " = " value %`

The third example of both blocks shows a comparison checking for a non-empty value or existence. This test can also be omitted.

- `%paramDefault ? " = " value : ""%`
- `%paramDefault ? " = " value %`

All of the above examples containing `paramDefault` are equivalent. If the parameter in scope had a default value of **10**, the output from each of them would normally be:

`= 10`

Notes:

- In a conditional substitution macro, any white space following `<macroName>` is ignored; if white space is required in the output, it should be included within the quoted substitution strings

Tagged Value macros are a special form of field substitution macros, which provide access to element tags and the corresponding Tagged Values.

Use 1: Direct Substitution

This form of the macro directly substitutes the value of the named tag into the output.

Structure: `%<macroName>: " <tagName> " %`

`<macroName>` can be one of:

- `attTag`
- `classTag`
- `connectorDestElemTag`
- `connectorDestTag`
- `connectorSourceElemTag`

- connectorSourceTag
- connectorTag
- linkAttTag
- linkTag
- opTag
- packageTag
- paramTag

This corresponds to the tags for attributes, Classes, operations, packages, parameters, connectors with both ends, elements at both ends of connectors and connectors including the attribute end.

`<tagName>` is a string representing the specific tag name.

Examples:

```
%opTag: " attribute" %
```

Use 2: Conditional Substitution

This form of the macro mimics the conditional substitution defined for field substitution macros.

```
Structure: %<macroName>: "<tagName>" ( == "<test>" ) ? <subTrue> ( : <subFalse> ) %
```

Where:

- `<macroName>` and `<tagName>` are as defined above
- (`<text>`) denotes that `<text>` is optional
- `<test>` is a string representing a possible value for the macro
- `<sub True>` and `<sub False>` can be a combination of quoted strings and the keyword value. Where the value is used, it gets replaced with the macro's value in the output.

Examples:

```
%opTag: " opI nline" ? " i nline" : "" %
%opTag: " opI nline" ? " i nline" %
%cl assTag: " unsafe" == " true" ? " unsafe" : "" %
%cl assTag: " unsafe" == " true" ? " unsafe" %
```

Tagged Value macros use the same naming convention as field substitution macros.

Function macros are a convenient way of manipulating and formatting various element data. Each function macro returns a result string. There are two primary ways to use the results of function macros:

- Direct substitution of the returned string into the output, such as: `%TO_LOWER(attName)%`
- Storing the returned string as part of a variable definition such as: `$name = %TO_LOWER(attName)%`

Function macros can take parameters, which can be passed to the macros as:

- String literals, enclosed within double quotation marks
- Direct substitution macros without the enclosing percent signs
- Variable references
- Numeric literals.

Multiple parameters are passed using a comma-separated list.

Function macros are named according to the All-Caps style, as in: `%CONVERT_SCOPE(opScope)%`

The available function macros are described below. Parameters are denoted by angle brackets, as in: `FUNCTION_NAME(<param>)`.

CONVERT_SCOPE(<umIScope>)

For use with supported languages. Converts <umIScope> to the appropriate scope keyword for the language being generated. The following table shows the conversion of <umIScope> with respect to the given language.

Language	Package	Public	Private	Protected
C++	public	public	private	protected
C#	internal	public	private	protected
Delphi	protected	public	private	protected
Java		public	private	protected
PHP	public	public	private	protected
VB	Protected	Public	Private	Protected
VB .Net	Friend	Public	Private	Protected

COLLECTION_CLASS(<language>)

Gives the appropriate collection Class for the language specified for the current linked attribute.

CSTYLE_COMMENT(<wrap_length>)

Converts the notes for the element currently in scope to plain C-style comments, using /* and */.

DELPHI_PROPERTIES(<scope>, <separator>, <indent>)

Generates a Delphi property.

DELPHI_COMMENT(<wrap_length>)

Converts the notes for the element currently in scope to Delphi comments.

EXEC_ADD_IN(<addin_name>, <function_name>, <prm_1>, ..., <prm_n>)

Invokes an Enterprise Architect Add-In function, which can return a result string. <addin_name> and <function_name> specify the names of the Add-In and function to be invoked. Parameters to the Add-In function can be specified via parameters <prm_1> to <prm_n>. For example:

```
$result = %EXEC_ADD_IN( " MyAddin" , " ProcessOperation" , classGUID , opGUID ) %
```

Any function that is to be called by the *EXEC_ADD_IN* macro must have two parameters: an *EA.Repository* object, and a *Variant* array that contains any additional parameters from the *EXEC_ADD_IN* call. Return type should be *Variant*. For example:

```
Public Function ProcessOperation( Repository As EA.Repository , args As Variant ) As Variant
```

FIND(<src>, <subString>)

Position of the first instance of <subString> in <src>; -1 if none.

GET_ALIGNMENT()

Returns a string where all of the text on the current line of output is converted into spaces and tabs.

JAVADOC_COMMENT(<wrap_length>)

Converts the notes for the element currently in scope to *javadoc*-style comments.

LEFT(<src>, <count>)

The first <count> characters of <src>.

LENGTH(<src>)

Length of <src>.

MID(<src>, <count>)**MID(<src>, <start>, <count>)**

Substring of <src> starting at <start> and including <count> characters. Where <count> is omitted the rest of the string is included.

PI(<option>, <value>, ...)

Sets the PI for the current template to <value>. <option> controls when the new PI takes effect. Valid values are:

- *I, Immediate*: the new PI is generated before the next non-empty template line
- *N, Next*: the new PI is generated after the next non-empty template line.

Multiple pairs of options are allowed in one call. For more details, see the [description of PI](#).^[1135]

PROCESS_END_OBJECT(<template_name>)

Enables the Classes that are one Class further away from the base Class, to be transformed into objects (such as attributes, operations, packages, parameters and columns) of the base Class. <template_name> refers to the working template that temporarily stores the data.

REMOVE_DUPLICATES(<source>, <separator>)

Where <source> is a <separator> separated list; this removes any duplicate or empty strings.

REPLACE(<string>, <old>, <new>)

Replaces all occurrences of <old> with <new> in the given string <string>.

RESOLVE_OP_NAME()

Resolves clashes in interface names where two method-from interfaces have the same name.

RESOLVE_QUALIFIED_TYPE()**RESOLVE_QUALIFIED_TYPE(<separator>)****RESOLVE_QUALIFIED_TYPE(<separator>, <default>)**

Generates a qualified type for the current attribute, linked attribute, linked parent, operation, or parameter. Enables the specification of a separator other than . and a default value for when some value is required.

RIGHT(<src>, <count>)

The last <count> characters of <src>.

TO_LOWER(<string>)

Converts <string> to lower case.

TO_UPPER(<string>)

Converts <string> to upper case.

TRIM(<string>)**TRIM(<string>, <trimChars>)**

Removes trailing and leading white spaces from <string>. If <trimChars> is specified, all leading and trailing characters in the set of <trimChars> are removed.

TRIM_LEFT(<string>)**TRIM_LEFT(<string>, <trimChars>)**

Removes the specified leading characters from <string>.

TRIM_RIGHT(<string>)**TRIM_RIGHT(<string>, <trimChars>)**

Removes the specified trailing characters from <string>.

VB_COMMENT(<wrap_length>)

Converts the notes for the element currently in scope to Visual Basic style comments.

WRAP_COMMENT(<comment>, <wrap_length>, <indent>, <start_string>)

Wraps the text <comment> at width <wrap_length> putting <indent> and <start_string> at the beginning of each line. For example:

```
$behavior = %WRAP_COMMENT( opBehavior, "40", " ", "//" ) %
```

<wrap_length> must still be passed as a string, even though WRAP_COMMENT treats this parameter as an integer.

WRAP_LINES(<text>, <wrap_length>, <start_string>(, <end_string>))

Wraps <text> as designated to be <wrap_length>, adding <start_string> to the beginning of every line and <end_string> to the end of the line if it is specified.

XML_COMMENT(<wrap_length>)

Converts the notes for the element currently in scope to XML-style comments.

Control macros are used to control the processing and formatting of the templates. The basic types of control macro include:

- The *list* macro, for generating multiple element features, such as attributes and operations
- The branching macros, which form *if-then-else* constructs to conditionally execute parts of a template
- The PI macro, which takes effect from the next non-empty line
- A PI [function macro](#)^[1130] that enables setting PI to a variable and adds the ability to set the PI that is generated before the next line
- The PI macro for formatting new lines in the output
- The synchronization macros.

In general, control macros are named according to Camel casing.

List

The *list* macro is used to generate multiple elements. The basic structure is:

```
%list = <TemplateName> @separator = <string> @indent = <string> ( <conditions>
```

) %

where *<string>* is a double-quoted literal string and *<TemplateName>* can be one of the following template names:

- Attribute
- Class
- ClassBase
- ClassImpl
- ClassInterface
- Constraint
- Custom Template (custom templates enable you to define your own templates; for more information see [Custom Templates](#))^[1147].
- Effort
- InnerClass
- InnerClassImpl
- LinkedFile
- Metric
- Namespace
- Operation
- OperationImpl
- Parameter
- Problem
- Requirement
- Resource
- Risk
- Scenario
- Test

<conditions> is optional and appears the same as the conditions for *if* and *elseif* statements.

Example:

```
%d i st =" A t t r i b u t e" @separator="\n" @indent="  " %
```

The *separator* attribute, denoted above by *@separator*, specifies the space that should be used between the list items. This excludes the last item in the list.

The *indent* attribute, denoted by *@indent*, specifies the space by which each line in the generated output should be indented.

The above example would output the result of processing the *Attribute* template, for each attribute element of the Class in scope. The resultant list would separate its items with a single new line and indent them two spaces respectively. If the Class in scope had any stereotyped attributes, they would be generated using the appropriately specialized template.

There are some special cases to consider when using the *list* macro:

- If the *Attribute* template is used as an argument to the list macro, this also generates attributes derived from associations by executing the appropriate *LinkedAttribute* template
- If the *ClassBase* template is used as an argument to the list macro, this also generates Class bases derived from links in the model by executing the appropriate *LinkedClassBase* template
- If the *ClassInterface* template is used as an argument to the list macro, this also generates Class bases derived from links in the model by executing the appropriate *LinkedClassInterface* template
- If *InnerClass* or *InnerClassImpl* is used as an argument to the list macro, these Classes are generated using the *Class* and *ClassImpl* templates respectively. These arguments tell Enterprise Architect that it should process the templates based on the inner Classes of the Class in scope.

Branching (if-then-else Constructs)

The CTF supports a limited form of branching through the following macros:

- *if*
- *elseif*
- *endif*
- *endTemplate*

The basic structure of the *if* and *elseif* macros is:

```
%i f <test> <operator> <test>%
```

where *<operator>* can be one of:

- ==
- !=

and *<test>* can be one of:

- a string literal, enclosed within double quotation marks
- a direct substitution macro, without the enclosing percent signs
- a variable reference.

Branches can be nested, and multiple conditions can be specified using one of:

- and
- or.

When specifying multiple conditions, *and* and *or* have the same order of precedence, and conditions are processed left to right.

The *endif* or *endTemplate* macros must be used to signify the end of a branch. In addition, the *endTemplate* macro causes the template to return immediately, if the corresponding branch is being executed.

Example:

```
%i f elemType == "Interface" %
;
%e l s e %
%O p e r a t i o n B o d y %
%e n d I f %
```

Example:

```
$bases=%d i s t="ClassBase" @separator=", "%
$I n t e r f a c e s=%d i s t="ClassInterface" @separator=", "%
%i f $bases != "" and $i n t e r f a c e s != "" %
: $bases, $i n t e r f a c e s
%e l s e l f $bases != "" %
: $bases
%e l s e l f $i n t e r f a c e s != "" %
: $i n t e r f a c e s
%e n d I f %
```

The PI Macro

There are two primary means of generating whitespace from the templates:

- Explicitly using the *newline*, *space* and *tab* characters (**\n**, **\t**) as part of Literal Text
- Using the *PI* macro to format lines in the template that result in non-empty substitutions in the output.

By default, each template line that generates a non-empty substitution also results in a newline being produced in the output. This behavior can be changed through the *PI* macro.

To demonstrate the use of the *PI* macro, consider the default *C# Operation* template:

```
%o p T a g: " A t t r i b u t e " %
```

Default PI is \n, so any attributes would be on their

```

%PI = " " %
%opTag: " unsafe" == " true" ?
" unsafe" : "" %
%CONVERT_SCOPE( opScope) %
%opTag: " new" == " true" ? " new" :
"" %
%opAbstract == " T" ? " abstract" :
"" %
%opConst == " T" ? " sealed" : "" %
%opStatic == " T" ? " static" : "" %
%opTag: " extern" == " true" ?
" extern" : "" %
%opTag: " delegate" == " true" ?
" delegate" : "" %
%opTag: " override" == " true" ?
" override" : "" %
%opTag: " virtual" == " true" ?
" virtual" : "" %
%opRetur nType%%opRetur nAr ray == " T"
? "( )" : "" %
%opSt er eot ype == " oper at or" ?
" oper at or" : "" %
%opName%( %d i st = " Par amet er "
@separ at or = " , " %)

```

own line

Blank lines have no effect on the output

Set the PI, so keywords are separated by a space

Any keyword that does not apply - that is, the macro produces an empty result - does not result in a space

Only one space is generated for this line

The final line in the template does not generate a space

In the above example macros for the various keywords are to be arranged vertically for readability. In the output, however, each relevant keyword is to be separated by a single space. This is achieved by the line:

```
%PI = " " %
```

Notice how you do not specify the space between each of the possible keywords. This space is already implied by setting the PI to a single space. Essentially the PI acts as a convenience mechanism for formatting the output from within the templates.

The structure for setting the processing instruction is:

```
%PI =<value>%
```

where <value> can be a literal string enclosed by double quotes.

The following points apply to the *PI* macro:

- The value of the PI is not accessed explicitly
- Only template lines that result in a non-empty substitution cause the PI to be generated
- The last non-empty template line does not cause the PI to be generated
- The PI is not appended to the last substitution, regardless of which template line caused that substitution.

Synchronization Macros

The *synchronization macros* are used to provide formatting hints to Enterprise Architect when inserting new sections into the source code, during forward synchronization. The values for synchronization macros must be set in the **File** templates.

The structure for setting synchronization macros is:

```
%<name>=<value>%
```

where <name> can be one of the macros listed below and <value> is a literal string enclosed by double quotes.

Macro Name	Description
synchNewClassNotesSpace	Space to append to a new Class note. Default value: \ n .
synchNewAttributeNotesSpace	Space to append to a new attribute note. Default value: \ n .
synchNewOperationNotesSpace	Space to append to a new operation note. Default value: \ n .
synchNewOperationBodySpace	Space to append to a new operation body. Default value: \ n .
synchNamespaceBodyIndent	Indent applied to Classes within non-global namespaces. Default value: \ t .

Enterprise Architect provides two Enterprise Architect Simulation Library (EASL) code generation macros to generate code from behavioral models. These are:

- EASL_GET and
- EASLList.

EASL_GET

The *EASL_GET* macro is used to retrieve a property or a collection of an EASL object. The EASL objects and the properties and collections for each object are identified in the [EASL Collections](#)^[1138] and [EASL Properties](#)^[1147] topics.

Syntax

```
$result = %EASL_GET( <<Property>>, <<Owner ID>>, <<Name>> )
```

where:

- <<Property>> is either "Property" or "Collection"
- <<OwnerID>> is the ID of the owner object for which the property/collection is to be retrieved
- <<Name>> is the name of the property or Collection being accessed
- \$result is the returned value; this is "" if not a valid property.

Example

```
$sPr opName = %EASL_GET( " Pr ope r t y" , $c o n t e x t , " N a m e" ) %
```

EASLList

The *EASLList* macro is used to render each object in an EASL collection using the appropriate template.

Syntax

```
$result = %EASLList = <<Template Name>> @separator = <<Separator>>
@indent = <<indent>> @owner = <<OwnedID>>
@collection = <<Collection Name>> @option1 = <<OPTI ON1>>
@option2 = <<OPTI ON2>> . . . . . @optionN = <<OPTI ONN>> %
```

where:

- <<TemplateName>> is the name of any [behavioral model template](#)^[1138] or [custom template](#)^[1147]
- <<Separator>> is a list separator (such as "\n")
- <<indent>> is any indentation to be applied to the result
- <<owner>> is the ID of the object that contains the required collection
- <<CollectionName>> is the name of the required collection

- <<OPTION1>>....<<OPTION99>> are miscellaneous options that might be passed on the template; each option is given as an additional input parameter to the template
- \$result is the resultant value; this is "" if not a valid collection.

Example

```
$sStates = %EASLList="State" @separator="\n" @indent="\t"
           @owner=$StateMachineGUID @collection="States"
           @option=$sOption%
```

Behavioral Model Templates

- Action
- Action Assignment
- Action Break
- Action Call
- Action Create
- Action Destroy
- Action If
- Action Loop
- Action Opaque
- Action Parallel
- Action RaiseEvent
- Action RaiseException
- Action Switch
- Behavior
- Behavior Body
- Behavior Declaration
- Behavior Parameter
- Call Argument
- Guard
- Property Object
- Property Declaration
- Property Notes
- State
- State CallBack
- State Enumerate
- State EnumeratedName
- StateMachine
- StateMachine HistoryVar
- Transition
- Transition Effect
- Trigger.

This topic lists the EASL collections for each of the EASL objects, as retrieved by the [EASL_GET](#)^[1137] code generation macro.

Action

Collection Name	Description
Arguments	The Action's arguments.
SubActions	The sub-actions of the Action.

Behavior

Collection Name	Description
Actions	The Behavior's Actions.
Nodes	The Behavior's nodes.
Parameters	The Behavior's parameters.
Variables	The Behavior's variables.

Classifier

Collection Name	Description
AllStateMachines	All State Machines for the Classifier.
AsynchProperties	The asynchronous properties of the Classifier.
AsynchTriggers	The asynchronous triggers of the Classifier.
Behaviors	The behaviors of the Classifier.
Properties	The properties of the Classifier.
TimedProperties	The timed properties of the Classifier.
TimedTriggers	The timed triggers of the Classifier.
Triggers	All triggers of the Classifier.

Construct

Collection Name	Description
AllChildren	The Construct's children.
ClientDependencies	The client dependencies on the Construct.
StereoTypes	The stereotypes of the Construct.
SupplierDependencies	The supplier dependencies on the Construct.

Node

Collection Name	Description
IncomingEdges	The Node's incoming edges.
OutgoingEdges	The Node's outgoing edges.
SubNodes	The sub-nodes of the Node.

State

Collection Name	Description
DoBehaviors	The State's Do behaviors.
EntryBehaviors	The State's Entry behaviors.
ExitBehaviors	The State's Exit behaviors.

StateMachine

Collection Name	Description
AllFinalStates	The State Machine's final States.
AllStates	All States within the State Machine, including those within Submachine States.
DerivedTransitions	The State Machine's derived transitions with the associated valid effect.
States	The States within the State Machine.
Transitions	The transitions within the State Machine.
Vertices	The State Machine's vertices.

Transition

Collection Name	Description
Effects	The Transition's effects.
Guards	The Transition's guards.
Triggers	The Transition's triggers.

Trigger

Collection Name	Description
TriggeredTransitions	The triggered transitions associated with the Trigger.

Vertex

Collection Name	Description
DerivedOutgoingTransitions	The Vertex's derived outgoing transitions after traversing the pseudo-nodes.
IncomingTransitions	The Vertex's incoming transitions.

Collection Name	Description
OutgoingTransitions	The Vertex's outgoing transitions.

This topic lists the EASL properties for each of the EASL objects, as retrieved by the [EASL_GET](#)^[1137] code generation macro.

Action

Property Name	Description
Behavior	The Action's associated behavior (<i>Call Behavior Action</i> or <i>Call Operation Action</i>).
Body	The Action's body.
Context	The Action's context.
Guard	The Action's guard.
IsFinal	A check on whether the action is a final Action.
IsGuarded	A check on whether the action is a guarded Action.
IsInitial	A check on whether the action is an initial Action.
Kind	The Action's kind.
Next	The Action's next action.
Node	The Action's associated node in the graph.

Argument

Property Name	Description
Parameter	The ID of the Argument's associated parameter.
Value	The default value of the argument.

Behavior

Property Name	Description
InitialAction	The Behavior's initial action.
isReadOnly	The isReadOnly of the Behavior.
isSingleExecution	The isSingleExecution of the Behavior.
Kind	The kind of Behavior.
ReturnType	The return type of the Behavior.
Specification	The specification of the Behavior.

CallEvent

Property Name	Description
Operation	The operation of the CallEvent.

ChangeEvent

Property Name	Description
ChangeExpression	The change expression of the ChangeEvent.

Classifier

Property Name	Description
HasBehaviors	A check on whether the Classifier has behavioral models (Activity and Interaction).
Language	The Classifier's language.
StateMachine	The State Machine of the Classifier.

Condition

Property Name	Description
Expression	The Condition's expression.
Lower	The Condition's lower value.
Upper	The Condition's upper value.

Construct

Property Name	Description
GetTaggedValue	The Property's Tagged Value.
IsStereotypeApplied	A check on whether a particular stereotype is applied to the Property.
Notes	Notes on the Property.
UMLType	The UML type of the Property.
Visibility	The visibility of the Property.

Edge

Property Name	Description
From	The ID of the node from which the Edge arises.
To	The ID of the node at which the Edge is targeted.

EventObject

Property Name	Description
EventKind	The event kind of the Event Object.

Instance

Property Name	Description
Classifier	The classifier of the Instance.
Value	The value of the Instance.

Parameter

Property Name	Description
Direction	The direction of the Parameter.
Type	The type of the Parameter.
Value	The value of the parameter.

Primitive

Property Name	Description
FQName	The FQ name of the Primitive.
ID	The ID of the Primitive.
Name	The name of the Primitive.
ObjectType	The object type of the Primitive.
Parent	The IDParent of the Primitive.

PropertyObject

Property Name	Description
BoundSize	The bound size of the PropertyObject (if it is a collection).

Property Name	Description
ClassifierStereoType	The stereotype of the PropertyObject's classifier.
IsAsynchProp	A check on whether the PropertyObject is an asynchronous property.
IsCollection	A check on whether the PropertyObject is a collection.
IsOrdered	A check on whether the PropertyObject is ordered (if it is a collection).
IsTimedProp	A check on whether the PropertyObject is a timed property.
Kind	The PropertyObject's kind.
LowerValue	The PropertyObject's lower value (if it is a collection).
Type	The PropertyObject's type.
UpperValue	The PropertyObject's upper value (if it is a collection).
Value	The PropertyObject's value.

SignalEvent

Property Name	Description
Signal	The signal of the SignalEvent.

State

Property Name	Description
HasSubMachine	A check on whether the State is a Submachine state.
IsFinalState	A check on whether the State is a final state.
SubMachine	Get the ID of the Submachine contained by the State (if applicable).

StateMachine

Property Name	Description
HasSubMachineState	A check on whether the State Machine has a Submachine state.
InitialState	The State Machine's initial state.
SubMachineState	The State Machine's Submachine state.

TimeEvent

Property Name	Description
When	The 'when' property of the TimeEvent.

Transition

Property Name	Description
HasEffect	A check on whether the transition has a valid effect.
IsDerived	A check on whether the transition is a derived transition.
IsTranscend	A check on whether the transition transcends from one State Machine (Submachine state) to another.
IsTriggered	A check on whether the transition is triggered.
Source	The Transition's source.
Target	The Transition's target.

Trigger

Property Name	Description
AsynchDestinationState	The asynchronous destination state of the Trigger (if it is an asynchronous trigger).
DependentProperty	The ID of the property associated with the Trigger.
Event	The Trigger's event.
Name	The Trigger's name.
Type	The Trigger's type.

Vertex

Property Name	Description
IsHistory	A check on whether the vertex is a history state.
IsPseudoState	A check on whether the vertex is a pseudo state.
PseudoStateKind	The Vertex's pseudo-state kind.

6.10.1.5.1.3 Variables

Template variables provide a convenient way of storing and retrieving data within a template. This section explains how variables are defined and referenced.

Variable Definitions

Variable definitions take the basic form:

```
$<name> = <value>
```

where *<name>* can be any alpha-numeric sequence and *<value>* is derived from a macro or another variable.

A simple example definition would be:

```
$foo = %className%
```

Variables can be defined, using values from:

- Substitution, function or list macros
- String literals, enclosed within double quotation marks
- Variable references.

Definition Rules

The following rules apply to variable definitions:

- Variables have global scope within the template in which they are defined and are not accessible to other templates
- Each variable must be defined at the start of a line, without any intervening whitespace
- Variables are denoted by prefixing the name with \$, as in \$foo
- Variables do not have to be declared, prior to being defined
- Variables must be defined using either the assignment operator (=), or the addition-assignment operator (+=)
- Multiple terms can be combined in a single definition using the addition operator (+).

Examples

Using a substitution macro:

```
$foo = %opTag: " bar " %
```

Using a literal string:

```
$foo = " bar "
```

Using another variable:

```
$foo = $bar
```

Using a list macro:

```
$ops = %list=" Operation" @separator="\n\n" @indent="\t" %
```

Using the addition-assignment operator (+=):

```
$body += %list=" Operation" @separator="\n\n" @indent="\t" %
```

The above definition is equivalent to the following:

```
$body = $body + %list=" Operation" @separator="\n\n" @indent="\t" %
```

Using multiple terms:

```
$templateArgs = %list=" ClassParameter" @separator=", " %
$template = "template<" + $templateArgs + ">"
```

Variable References

Variable values can be retrieved by using a reference of the form:

```
$<name>
```

where <name> can be a previously defined variable.

Variable references can be used in one of the following ways:

- As part of a macro, such as the argument to a function macro
- As a term in a variable definition
- As a direct substitution of the variable value into the output.

It is legal to reference a variable before it is defined. In this case, the variable is assumed to contain an empty string value: ""

Example 1

Using variables as part of a macro. The following is an excerpt from the default C++ ClassNotes template.

<code>\$wrapLen = %genOpt WrapComment %</code>	Define variables to store the style and wrap length options.
<code>\$style = %</code>	
<code>genOpt CPPComment Style%</code>	
<code>%if \$style == "XML.NET" %</code>	Reference to <i>\$style</i> as part of a condition.
<code>%XML_COMMENT(\$wrapLen) %</code>	
<code>%else%</code>	
<code>%CSTYLE_COMMENT(\$wrapLen) %</code>	Reference to <i>\$wrapLen</i> as an argument to function macro.
<code>%endif %</code>	

Example 2

Using variable references as part of a variable definitions:

<code>\$foo = "foo"</code>	Define our variables.
<code>\$bar = "bar"</code>	
<code>\$foobar = \$foo + \$bar</code>	<i>\$foobar</i> now contains the value <i>foobar</i> .

Example 3

Substituting variable values into the output

<code>\$bases=%classInherits%</code>	Store the result of the <i>ClassInherits</i> template in <i>\$bases</i> .
<code>Class %className%\$bases</code>	Now output the value of <i>\$bases</i> after the Class name.

6.10.1.5.2 The Code Template Editor in MDG Development

The following topics describe how you use the Code Template Editor window to create custom templates:

- [Custom Templates](#) ^[1147]
- [Override Default Templates](#) ^[1148]
- [Add New Stereotyped Templates](#) ^[1149]
- [Create Templates For Custom Languages](#) ^[1150]

The Code Template Editor provides the facilities of the **Common Code Editor**, including intellisense for the code generation template macros. For more information on intellisense and the Common Code Editor, see the Code Editors topic.

Learn More:

- [Code Generation Template Macros](#) ^[1151]
- [Code Editors](#) ^[1394]

6.10.1.5.2.1 Custom Templates

Custom templates enable you to generate an element in many different ways. Enterprise Architect enables you to define custom templates that are associated with given elements and call these templates from existing templates. You can even add stereotype overrides to your custom templates. For example, you might list all of your parameters and their notes in your method notes.

To create a new custom template, follow the steps below:

Step	Description	See Also
1	Select the Settings Code Generation Templates menu option, or press (Ctrl+Shift+P). The Code Templates Editor tab opens.	
2	In the Language field, click on the drop-down arrow and select the appropriate language.	
3	Click on the Add New Custom Template button. The Create New Custom Template dialog displays.	
4	<p>In the Template Type field, click on the drop-down arrow and select the appropriate element. The elements currently supported are:</p> <ul style="list-style-type: none"> • Attribute • Class • Class Base • Class Interface • Class Parameter • Connector • Import • Linked Attribute • Linked Class Base • Linked Class Interface • Namespace • Operation • Parameter <p><None> requires special treatment. It enables the definition of a function macro that doesn't actually apply to any of the types, but must be called as a function to define variables \$parameter1, \$parameter2 and so on for each value passed in.</p>	
5	In the Template Name field, type an appropriate name, then click on the OK button.	
6	On the Code Templates Editor tab, the new template displays in the Templates list with the value Yes in the Modified field. The template is called <Template Type>_<Template Name>.	
7	Select the appropriate template from the Templates list and edit the contents in the Template field to meet your requirements.	
8	Click on the Save button. This stores the new stereotyped template in the .EAP file. The template is now available from the list of templates and via direct substitution for use.	

6.10.1.5.2.2 Override Default Templates

Enterprise Architect has a set of built-in or default code generation templates. The **Code Templates Editor** enables you to modify these default templates, hence customizing the way in which Enterprise Architect generates code. You can choose to modify any or all of the base templates to achieve your required coding style.

Any templates that you have overridden are stored in the .EAP file. When generating code, Enterprise Architect first checks whether a template has been modified and if so, uses that template. Otherwise the appropriate default template is used.

To override a default code generation template, follow the steps below

Step	Description	See Also
1	Select the Configuration Code Generation Templates menu option. The Code Templates Editor displays.	
2	Select the appropriate language from the Language list.	
3	Select one of the base templates from the Templates list.	
4	If the base template has stereotyped overrides, you can select one of these from the Stereotype Overrides list.	
5	In the Code Templates Editor , make the required modifications.	
6	Click on the Save button. This stores the modified version of the template to the .EAP file. The template is marked as modified.	

When generating code, Enterprise Architect now uses the overridden template, instead of the default template.

6.10.1.5.2.3 Add New Stereotyped Templates

Sometimes it is useful to define a specific code generation template for use with elements of a given stereotype. This enables different code to be generated for elements, depending on their stereotype. Enterprise Architect provides some default templates, which have been specialized for commonly used stereotypes in supported languages. For example the Operation Body template for C# has been specialized for the *property* stereotype, so that it automatically generates its constituent *get* and *set* methods. Users can override the default stereotyped templates as described in the previous topic. Additionally users can define templates for their own stereotypes, as described below.

To add a new stereotyped template, follow the steps below:

Step	Description	See Also
1	Select the Configuration Code Generation Templates menu option to open the Code Templates Editor .	
2	Select the appropriate language, from the Language list.	
3	Select one of the base templates, from the Templates list.	
4	Click on the Add New Stereotyped Override button. The New Template Override dialog displays.	
5	Select the required Feature and/or Class stereotype and click on the OK button.	
6	The new stereotyped template override displays in Stereotype Overrides list, marked as modified.	
7	Make the required modifications in the Code Templates Editor .	
8	Click on the Save button. This stores the new stereotyped template in the .EAP file.	

Enterprise Architect can now use the stereotyped template, when generating code for elements of that stereotype.

Note that Class and feature stereotypes can be combined to provide a further level of specialization for features. For example, if properties should be generated differently when the Class has a stereotype

MyStereotype, then both *property* and *MyStereotype* should be specified in the **New Template Override** dialog.

6.10.1.5.2.4 Create Custom Language Template

Enterprise Architect can forward generate code for languages that it does not specifically support, if the appropriate code generation templates are defined for that language. This topic outlines the steps required to define templates for custom languages.

To define a template for a custom language, follow the steps below

Step	Description	See Also
1	Create the custom language as a new product. To do this: <ul style="list-style-type: none"> Select the Settings Code Datatypes menu option. The Programming Languages Datatypes dialog displays. In the Product Name field type the name of the new language, and in the Datatype field type a datatype (one is enough to declare that the new language exists). 	Data Types 777
2	Select the Settings Code Generation Templates menu option. The Code Templates Editor view displays.	
3	In the Language field, click on the drop-down arrow and select the custom language.	
4	From the Templates list, select one of the base templates.	
5	Define the template using the Code Templates Editor .	
6	Click on the Save button. This stores the template in the .EAP file.	
7	Repeat steps 1 to 6 for each of the relevant base templates for the custom language.	

Notes:

- The File template must be defined for the custom language. The File template can then see the Import Section, Namespace and Class templates

Part



7 Requirement Models

Topics:

Topic	Detail	See also
Introduction	<p>This section describes the Enterprise Architect Requirements Management facilities, and discusses:</p> <ul style="list-style-type: none"> • What requirements are • How requirements are generated and organized • How Enterprise Architect supports and simplifies Requirements Management 	White Paper (Requirements Management with Enterprise Architect)
What is a Requirement?	<p>Requirements are essentially what the system, application or business process <i>is required</i> to do</p> <p>A requirement can be:</p> <ul style="list-style-type: none"> • Broad and high level, defining - for example - that a process is necessary to update a particular database • More specialized and detailed, recording the expectation that - for example - a system call must always be acknowledged by return <p>Detailed requirements can be organized into a hierarchy culminating in a high-level requirement, so that satisfying each of the detailed requirements results in meeting the higher-level requirements and ultimately the top-level requirement</p> <p>This hierarchical structure helps manage the complexity of large systems with thousands of requirements and many processes being developed to implement the requirements</p>	
Gathering Requirements	<p>Gathering requirements is typically the first step in developing a solution, be it for developing a system or a process. Requirements are gathered from all parties expected to use, maintain or benefit from the solution, and are organized into groups, functional areas and hierarchies as necessary. They can be transcribed into a spreadsheet or a requirements gathering or management tool, or they can be created in an integrated modeling tool such as Enterprise Architect.</p>	
Requirements Management and Enterprise Architect	<p>The management of requirements is one of the more problematic disciplines in software development, for reasons such as:</p> <ul style="list-style-type: none"> • Diverse group input into the requirements • Organizational boundary divisions • Tool boundary divisions • Volatility of requirements • Imprecision and ambiguity in natural languages <p>These can cause issues with:</p> <ul style="list-style-type: none"> • Traceability and • Integration with change and configuration management systems <p>Enterprise Architect can reduce or eliminate these</p>	Create Requirements ^[1156] View Requirements ^[1170] Import Requirements Using CSV ^[1162] Model Requirements ^[1165] Requirement Properties ^[1157] Extend Requirement Properties ^[1160] Trace Use Of Requirements ^[1170] Relationship Matrix ^[496]

Topic	Detail	See also
	<p>problems in Requirements Management; it is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model. Within Enterprise Architect, you can:</p> <ul style="list-style-type: none"> • Create and view requirements as entities and properties directly in the model • Collate the requirements in an external CSV file and then import them into your model • Detail use cases and scenarios directly in the model • Enter standard attributes (properties) for each requirement, such as difficulty, status and type, and define your own attributes (properties) • Trace requirements to Use Cases, business rules, test cases and analysis artifacts (using, for example, the Relationship Matrix) • Trace and view the impact of changes on requirements (through, for example, the Traceability window) and review the changes themselves • Create customer-quality MS Word and HTML reports on requirements 	<p>Traceability Window^[495]</p> <p>Manage Requirement Changes^[1171]</p> <p>Report on Requirements^[1173]</p>

Notes:

- All of these features are illustrated by examples in the EAExample.eap model, provided as part of your Enterprise Architect installation in the Enterprise Architect Program Files directory:

..\Program Files\Sparx Systems\EA

7.1 Requirements

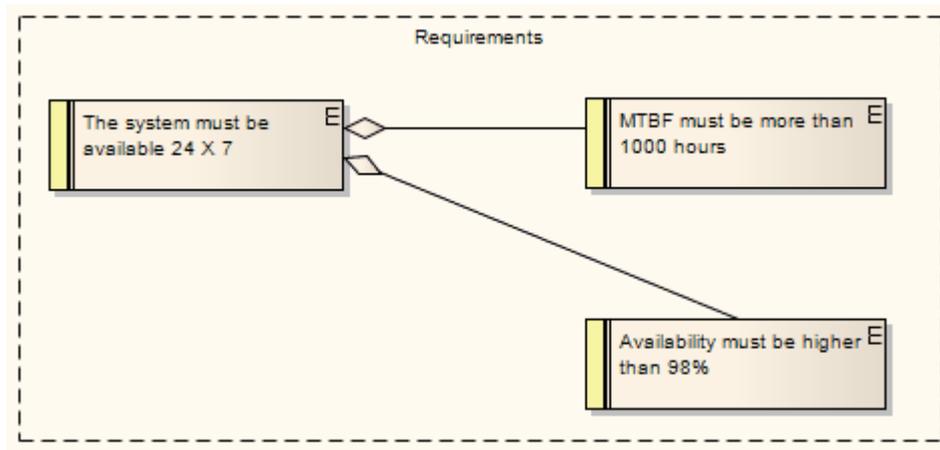
Description

As an analysis step, often it is desirable to capture simple system requirements. These are eventually realized by Use Cases.

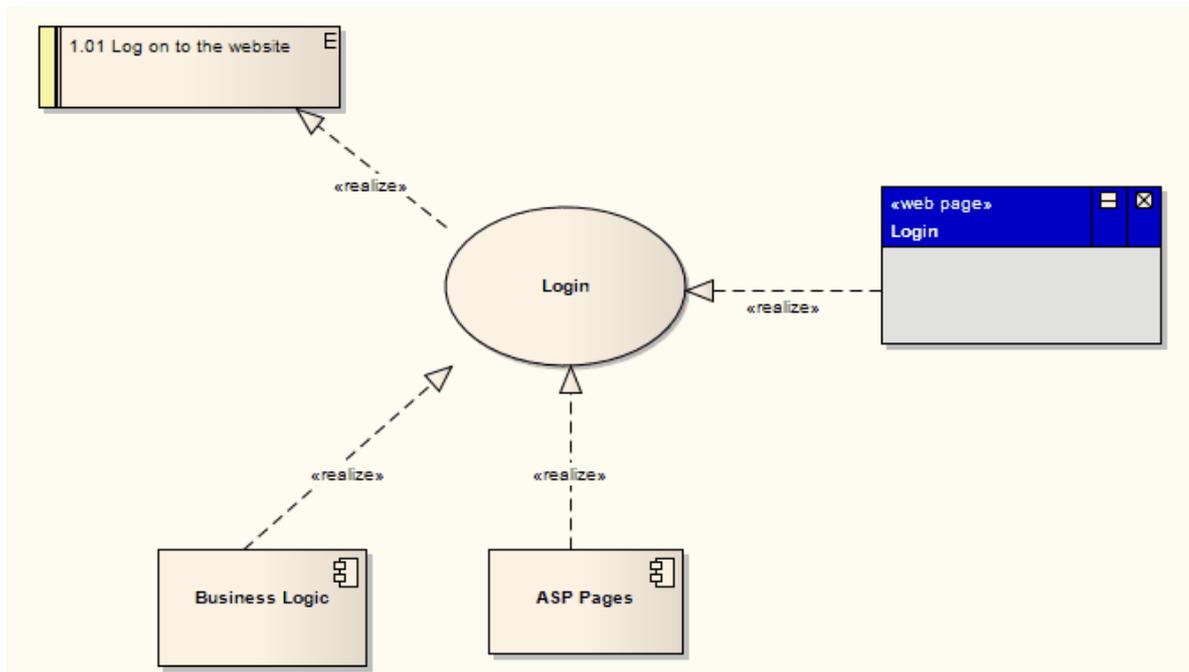
In the initial requirement gathering phase, cataloging requirements can be achieved using the Requirement extension on a Custom diagram.

Examples

Requirements can be aggregated to create a hierarchy. The diagram below illustrates how this might be done.



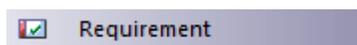
In the following diagram, a requirement that a user can log into a website is implemented by the *Login* Use Case, which in turn is implemented by the *Business Logic*, *ASP Pages* and *Login Web Page*. Using this approach, you can easily model quite detailed and complex dependencies and implementation relationships.



Notes:

- External requirements can be displayed with or without an identifying **E** (for External) in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the **Options** dialog, **Objects** page.
- The colors on Requirement elements identify the status of the requirement. You change the status - and hence color - on the element **Properties** dialog. You set the color for each status on the **Status Types** dialog.

Toolbox Icon



7.2 Create Requirements

Topics:

Topic	Detail	See also
Usage	<p>Within Enterprise Architect you can create external Requirement elements in a number of ways, such as:</p> <ul style="list-style-type: none"> • Dragging a <i>Requirement</i> icon from the Toolbox into a specific diagram (which also adds the Requirement to the diagram's parent package - see below) • Dragging text from a text file onto a diagram, to generate a requirement based on that text - see below • Generating an element within a specific package in the Project Browser - see below • Importing requirements from a spreadsheet application such as Excel, via CSV • Creating Requirement elements on the Package Browser or Diagram List for the selected package or diagram • Converting an internal responsibility into an external element, in a selected target package • Importing requirements from another requirements management tool, such as Telelogic DOORS (in this case via the Sparx Systems MDG Link For DOORS integration tool) 	<p>Import Requirements using CSV^[1162]</p> <p>Package Browser^[456]</p> <p>Diagram List^[462]</p> <p>Converting an Internal Responsibility^[1167]</p> <p>Sparx Systems MDG Link For DOORS</p>

How To:

To create Requirement elements in a diagram, follow the steps below:

Step	Action	See Also
1	Open the Custom pages on the Toolbox	Custom Group ^[562]
2	Drag the <i>Requirement</i> element onto the current diagram	
3	Enter the Name and other details for the requirement Enterprise Architect creates a Requirement element in the current diagram and in the diagram's parent package	Requirement Properties ^[1157]

Alternatively:

(The following procedure converts a text section heading into an element name and the section text into the element's Notes text. You can use this procedure to generate elements of a *range* of types; however, it is particularly useful for generating Requirements from a requirements specification document.)

Step	Action	See Also
1	Open an Enterprise Architect diagram in the Diagram View	
2	Open the required document file containing the text you want to generate Requirement elements from (this can be opened in any common text editing	

Step	Action	See Also
	tool)	
3	Highlight the required heading and associated text and drag them from the text file into the diagram The Toolbox Shortcut menu displays	Toolbox Shortcut Menu ^[55]
4	Navigate through the menus and select the required element type (in this case, click on Common and Requirement)	
5	Enterprise Architect creates a (Requirement) element in the diagram, and displays the Properties dialog with the section heading in the Name (or equivalent) field and the text in the Notes field; the element is also added to the diagram's parent package	

To create a requirement directly in the Project Browser, follow the steps below

Step	Action	See Also
1	Right-click on the required parent package to open the context menu	
2	Select the Insert New Element menu option; alternatively, press (Ctrl+M)	
3	In the New Element dialog select the Requirement type	
4	Enter the Name (or select Auto) and click on the OK button Enterprise Architect creates a requirement in the current package	

Notes:

- The Requirement element name can be simply descriptive text, with or without a manually-typed reference number; however, as requirements often have to have a unique reference for external checking, you can use the Enterprise Architect auto-numbering facility to automatically apply a numbering system with or without prefixes and suffixes - set the element type to **Requirement**
- External Requirement elements can be displayed with or without an identifying **E** in the top right corner; to toggle display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page
- Requirement elements can be color coded to indicate their status

Learn More:

- [Auto Naming and Auto Counters](#) ^[628]
- [Objects](#) ^[432]
- [Color Code External Requirements](#) ^[1159]

7.2.1 Requirement Properties

Requirement properties differ slightly from the properties of other elements; they include information related to the Type, Status, Difficulty and Priority of the Requirement. The **Notes** field is also important, as it describes precisely what requirement the element represents. Requirement naming can also require careful consideration and could reflect either a categorical naming convention, or simply a loose English description of the Requirement.

Access: [Requirement Element Context Menu | Properties](#)

Use to:

- Document requirements
- Set Requirement features such as Type, Status and Priority
- Set other element properties common to both Requirements and other model Elements

Reference:

Field	Usage	See Also
Short Description	The name of this requirement, which could include numbering, plain English description or some other formal specification	
Alias	An alias to be used for this requirement	Auto Counters ^[628]
Status	The current status of this requirement	Color Code Requirements ^[1159]
Difficulty	An estimate of the difficulty in meeting this requirement. Enterprise Architect supports the values 'Low', 'Medium', 'High'	
Priority	The relative importance of meeting this requirement compared to other requirements; select from: <ul style="list-style-type: none"> • Low • Medium • High 	
Author	The author of this requirement	Authors ^[778]
Key Words	A set of words that could be used to index or define the subject of this requirement	
Type	The type of this requirement. This is typically used as a category for the requirement.	Requirement Types ^[785]
Phase	The phase of this requirement.	
Version	The version of this requirement.	
Last Update	Read-only field specifying when this requirement was last changed.	
Created	Read-only field specifying when this requirement was first created.	
Notes	The description of this requirement.	Note Editors ^[769] , Linked Documents ^[729]

Notes:

- In Requirement Management tools and texts, the characteristics of a requirement are commonly called attributes. However, in UML the term attribute refers to a different type of feature, and the requirement characteristics are defined as properties. In this Enterprise Architect documentation, the term

properties is used

- In a project, it might be necessary to define more information in a requirement than is provided by the standard properties. For more information on extending the requirement properties, see the Extend Requirement Properties topic

Learn More

- [Extend Requirement Properties](#) ^[1160]

7.2.2 Color Code External Requirements

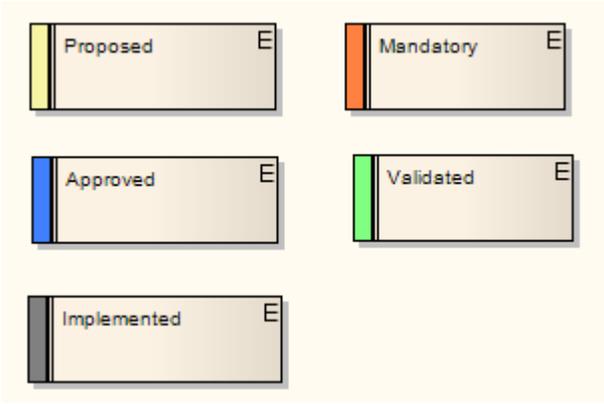
External requirements can be color coded to provide quick visual cues indicating the status of a requirement.

How to:

To enable color coded external requirements follow the steps below

Step	Action
1	Select the Tools Options menu option The Options dialog displays
2	From the hierarchical tree select Objects, and select the Show status colors on diagrams checkbox to enable the status of external requirements to be represented by color coding

Topics:

Topic	Detail	See also
Usage	<p>The color code requirements use the following default conventions:</p> <ul style="list-style-type: none"> • Yellow for Proposed • Blue for Approved • Green for Validated • Orange for Mandatory • Black for Implemented  <p>You can change these colors, and add or remove status types, using the Status Types dialog</p>	<p>Status Types Dialog ^[783]</p>

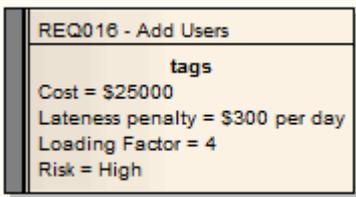
7.2.3 Extend Requirement Properties

Topics:

Topic	Detail	See also
Introduction	<p>A project might apply further properties to a requirement, such as cost, lateness penalty or risk to the business if not met. You can add these properties to specific Requirement elements, or configure them to be automatically available in all Requirement elements on creation, using Tagged Values. (These are sometimes referred to as User-defined attributes.)</p> <p>Extended element properties are not visible unless you open the Tagged Values window for the element. Alternatively, you can display the additional properties on the element image on its diagrams.</p>	<p>Tagged Values ^[762]</p> <p>Display Tagged Values on Diagrams ^[1167]</p>
Add Tagged Values to Existing Requirements	<p>To add a property to a single Requirement as a Tagged Value, simply click on the Requirement, display the Tagged Values window (Ctrl+Shift+6), and enter the name of the property as the tag name and the value of the property as the tag value.</p> <p>It is likely that any property you add to one Requirement would also apply to others. You might therefore use a predefined Tagged Value Type to identify your Requirement property, so that you can select it whenever required. The predefined Tagged Value Type also enables you to define specific values for the Tagged Value. If the appropriate predefined Tagged Value Type does not exist, a Technology Developer can create it to add to the structured tags, reference tags, or customized tags collections.</p>	<p>Assign a Tagged Value to an Item ^[764]</p> <p>Create Structured Tagged Values ^[1117]</p> <p>Create Reference Date Tagged Values ^[1113]</p> <p>Create a Custom Tagged Value Type ^[1113]</p>
Configure Requirements to be Created With Extended Properties	<p>If it is necessary to create all Requirements with the same extended set of properties, you can create a Requirement Template diagram and either create a special Requirement that defines those properties (as Tagged Values), or drag an existing Requirement with those properties onto the diagram. You then set the Requirement Template diagram as the template for all new Requirement elements, so that those new Requirements automatically have all of the properties you want.</p> <p>However, this then excludes other Requirement element formats, including the standard Requirement format. If you want to use another Requirement format, you have to replace or cancel the current Template. Alternatively, you can create a Profile.</p> <p>A Profile also defines exactly what a new Requirement element should contain, and how it should display in diagrams. However, a Profile is a collection of alternative element definitions, so it does not override the default Requirement format, nor does it prevent you from defining several different types of Requirement element. You can therefore have separate and parallel definitions of elements for business requirements, system requirements, project requirements, or any other category of requirement you decide to work with.</p> <p>For information on importing and using existing Profile files, see the UML Profiles topic. For information on creating new Profiles, see the Developing Profiles topic.</p>	<p>Set Element Template Package ^[643]</p> <p>UML Profiles ^[1026]</p> <p>Developing Profiles ^[1039]</p>

7.2.4 Display Tagged Values On Diagrams

Topics:

Topic	Detail	See also
Usage	<p>If you have extended the properties of a Requirement, you might want to make those properties visible in the Requirement elements in your diagrams, by switching on display of the element tags compartment. You can do this in one of two ways:</p> <ul style="list-style-type: none"> To display the <i>tags</i> compartment on all elements on a diagram, double-click on the diagram background and select the Elements tab of the Diagram Properties dialog; select the Tags checkbox and click on the OK button To display the <i>tags</i> compartment on a specific element on a diagram, right-click on the element and select the Feature Visibility context menu option; select the Tags checkbox in the Show Element Compartments panel of the Feature Visibility dialog, and click on the OK button <p>The Tagged Values are then displayed in the Requirement element on the diagram.</p> 	

Learn More:

- [Elements](#)^[573]
- [Feature Visibility](#)^[585]

7.2.5 Connect Requirements

Topics:

Topic	Detail	See also
Abstract	<p>A Requirement element can be connected to other Requirements, most commonly using Aggregate relationships to form a hierarchy of requirements.</p> <p>Requirements are also connected to other types of element, most commonly Use Cases, by Realize or Implements relationships.</p> <p>These relationships are very important, both in identifying how the Requirements are organized and used in the model, and in tracing the development from the Requirements throughout the model. Both of these tasks are very simple in Enterprise Architect, because once a connector on a Requirement exists, Enterprise Architect automatically lists the Requirement in the:</p>	<p>Aggregate relationships^[968]</p> <p>Realization^[1007]</p> <p>Trace Requirements^[1170]</p> <p>Traceability^[495]</p> <p>Requirements^[664]</p> <p>Requirement</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> Requirements Traceability window (an important tool in examining the role of Requirements in the model) Requirements tab of the target element Properties dialog Links tab of the Requirement element Properties dialog Scenarios & Requirements window Relationships Window Dependency and Implementation reports Standard RTF output <p>The connector itself is also listed in the Links tab of the target element Properties dialog, and in the Relationship Matrix. There are, therefore, many ways to locate, view and track Requirement relationships.</p>	Properties ^[1157] Scenarios & Requirements ^[689] Relationships Window ^[504] Dependency ^[1783] Implementation reports ^[1785] Standard RTF output ^[1726] Links ^[666] Relationship Matrix ^[496]
Connect On Diagram	<p>Relationships can be created on a diagram by clicking on the appropriate connector icon from the Requirement and Common pages of the Toolbox, clicking on the source (originating) element, and dragging to the target element.</p> <p>If you are connecting elements in different packages, you can drag elements from the Project Browser onto a common diagram and set up the relationships there. Alternatively, you can quickly generate a Realize connector by dragging an existing Requirement element from the Project Browser over the element that implements the Requirement. Enterprise Architect interprets this as a request to create the Realize connector and does so automatically. The Requirement element is not added to the diagram. However, if you subsequently drag the Requirement onto the diagram the connector is already in place.</p>	Requirement Group ^[563] Common Group ^[552]
Connect Off Diagram	<p>You can also connect a Requirement element to other elements without necessarily having the elements on the same diagram, or having a diagram open.</p> <p>Use the Relationship Matrix to create relationships for requirements; this is a convenient way of quickly building up complex relationships and hierarchies.</p>	Creating Relationships ^[500]

7.2.6 Import Requirements and Hierarchies in CSV

Topics:

Topic	Detail	See also
Usage	<p>You can import Requirements from a spreadsheet application in CSV format. Before doing this you must create a CSV import file specification that:</p> <ul style="list-style-type: none"> In the Default Types field has the value requirement,package to import requirements and a package structure to contain them Has the Preserve Hierarchy checkbox selected Identifies the data fields on the spreadsheet that are to be 	

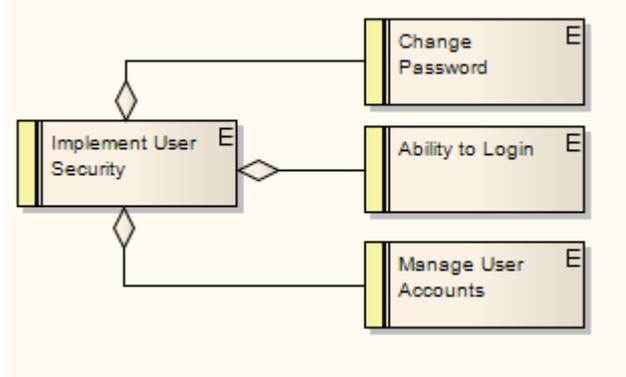
Topic	Detail	See also
	<p>translated into Enterprise Architect, in the order in which they are plotted across the spreadsheet</p> <ul style="list-style-type: none">• Is to operate on a spreadsheet containing the CSV_KEY and CSV_PARENT_KEY fields (which, if not generated by a CSV export from Enterprise Architect, you have added and populated yourself) <p>This enables you to import the individual and grouped requirements from the spreadsheet into Enterprise Architect, and to reconstruct the hierarchies of Requirements in the target package in the Project Browser.</p>	

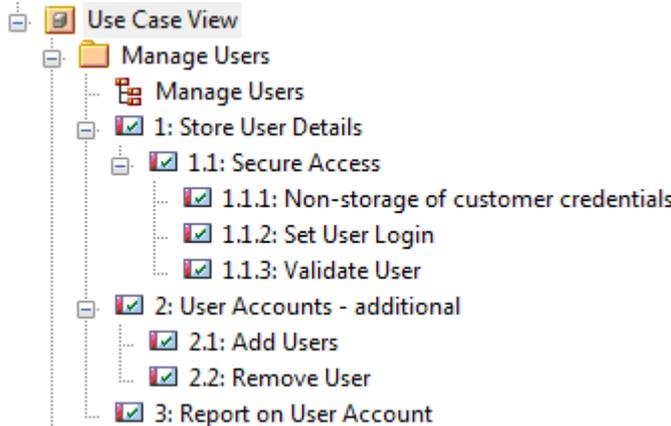
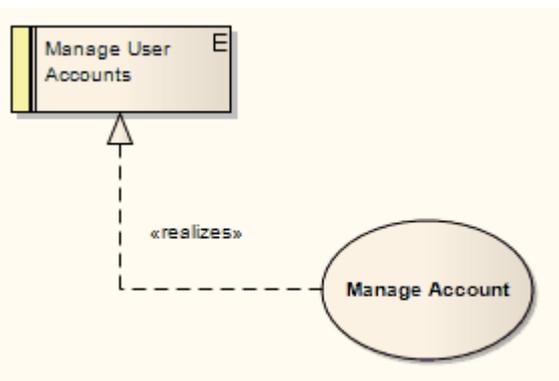
Learn More:

- [CSV Import](#) ^[342]
- [CSV Specifications](#) ^[338]
- [Using Preserve Hierarchy](#) ^[340]

7.3 Model Requirements

Topics:

Topic	Detail	See also
Represent Requirements	<p>In Enterprise Architect, a requirement can be modeled as an:</p> <ul style="list-style-type: none"> • External Requirement - an expectation of the system or process, what the system or process must provide, modeled as an element; for example, a business requirement or a stakeholder request - Requirements at this level have their own properties and are reported on separately in RTF reports • Internal requirement – a responsibility of an existing element, what the element must do or accomplish, defined as a property of the element <p>Requirements Management in Enterprise Architect is primarily concerned with external Requirement elements and the elements that implement or realize them.</p>	Requirements <small>[1154]</small> Internal requirements <small>[1167]</small>
Requirements in the Model	<p>Requirement elements can be grouped and organized within Requirements diagrams. The Requirement elements are connected to each other by <i>Aggregate</i> relationships to form a hierarchy:</p>  <p>It is quite usual to develop a package of many hundreds of Requirement elements, arranged individually and in hierarchies of varying complexity. In the Project Browser you can use the Turn On Level Numbering facility to highlight the order and arrangement of the Requirements quickly and easily. The following illustration shows a number of Requirements in a package, where Level Numbering makes the order and arrangement clear:</p>	Requirements Diagram <small>[1174]</small> Turn On Level Numbering <small>[444]</small> RTF Style Template Editor <small>[1736]</small>

Topic	Detail	See also
	 <p>If elements are added, moved or deleted from the package, the numbering automatically adjusts.</p> <p>This numbering can also be applied in the RTF report generator using the LevelNumber field in the Element section – {Element.LevelNumber}.</p>	
<p>Use Cases</p>	<p>Requirements are implemented (realized) by model elements such as Use Cases, Classes, Interfaces and Components. There are many ways to trace either the Requirement for the feature or service modeled by the elements, or the elements that develop the requirement, most visibly in Traceability diagrams that depict the Requirements and the model elements connected by Realize relationships. The Realize connector enables members of the project team to keep design objectives and development in tandem, and the development path and purpose clear.</p>  <p>The more usual realization relationship is between a Requirement and a Use Case. A Requirement can be realized by one or more Use Cases, and a Use Case can realize one or more Requirements.</p> <p>Whilst a Requirement defines a condition that must be met, the Use Case is the key to defining and visualizing how that condition is met. A Use Case diagram depicts the logical grouping of actions, processes and components to achieve a required result, and through the use of Actor elements also defines the user and/or system roles participating in the process.</p> <p>Each Use Case (as a composite element) can contain a combination of child diagrams that define in greater detail how a particular activity or facility might be implemented - such diagrams include Sequence,</p>	<p>Trace^[493]</p> <p>Traceability^[503]</p> <p>Connector Requirement^[1167]</p> <p>Use Case^[933]</p> <p>Use Case Diagram^[813]</p> <p>Actor^[877]</p> <p>Composite Element^[647]</p> <p>Sequence Diagram^[849]</p> <p>Communication Diagram^[859]</p> <p>Activity Diagram^[817]</p> <p>State Machine Diagram^[815]</p> <p>Business Rule Flow^[1205]</p>

Topic	Detail	See also
	<p>Communication, Activity, State Machine and Business Rule Flow diagrams. The actual implementation of each Use Case is realized by Class, Component and Interface elements organized in their own diagrams. These realizations can also be captured and viewed in Traceability diagrams, depicting the full development pathway from initial requirement through to testing and production.</p>	

7.3.1 Feature



Description

A *Feature* is a small, granular function or characteristic expressed in client-valued terms as a satisfaction of a requirement; for example: 'context-sensitive Help', or 'ability to reverse-engineer VB.Net'.

Features are the primary requirements-gathering artifact of the Feature-Driven Design (FDD) methodology. They define the product feature that satisfies what a Requirement element has formalized as a contractual, testable, expected deliverable (for example: requirement - 'every element must provide context-sensitive Help'; feature - 'every element provides context-sensitive Help'). One Feature might realize one or more Requirements, and one Requirement might be realized by more than one Feature.

Features also have relationships with Use Cases. A Use Case defines the interaction a user has with the system in order to satisfy one or more Requirements. The Feature identifies the facility that provides the means for that interaction.

Feature elements are non-UML and are not related to the UML elements of the same name which are either BehavioralFeatures (operations, or methods) or StructuralFeatures (Ports, Parts and attributes).

Feature elements are available from the **Requirements** page of the **Toolbox**.

Feature elements can be displayed with or without an identifying **F** in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the **Options** dialog, **Objects** page.

Toolbox Icon



Learn More:

- [Feature-Driven Design \(FDD\) methodology](#)
- [Requirement Element^{\[1153\]}](#)
- [Use Case Element^{\[935\]}](#)

7.3.2 Internal Requirements

Topic	Detail	See also
Usage	<p>Internal requirements in Enterprise Architect are element responsibilities. They are defined on the Requirements page of the element Properties dialog.</p> <p>Internal requirements form the functional requirements of the system to be built. The meaning of the requirement can vary depending on which element is the host; for example, a business process requirement might mean something different to a Use Case requirement, which again might mean something different to a Class requirement. In the example above, an internal responsibility to enable the user to login to the system has been defined for the Login Use Case. This is a responsibility of the Use Case - an action it is responsible for carrying out - and it applies only to this Use Case.</p> <p>The significant parameters (or, in Requirement Management terms, <i>attributes</i>) are the Type, Status, Difficulty and Priority. Whilst you can provide a detailed description of the responsibility in the Notes field, there is more scope in the name (Requirement field) to define the nature of the responsibility. An additional field, Stability, indicates the probability of the requirement changing; high stability means a low probability of change.</p> <p>The example Use Case above also has connections to two external requirements, which are system functions that the Use Case implements either in full or in part. You can convert an internal responsibility into an external requirement.</p> <p>You can also create internal responsibilities for an element using the Scenarios & Requirements window. A responsibility created in the window displays in the element Properties dialog, and vice versa.</p>	<p>Requirements ^[664]</p> <p>Make Internal Requirements External ^[1167]</p> <p>Scenarios & Requirements ^[689]</p>

7.3.2.1 Make Internal Requirement External

Elements in Enterprise Architect have internal requirements, or responsibilities (what they must do or accomplish). These often overlap or duplicate more formal requirements that the system in general must meet. You can move internal requirements to external requirements (where the requirement can perhaps be implemented by multiple elements) in one go, using the **Move External** function.

How To:

To change an internal requirement for an element into an external element, follow the steps below

Step	Action	See Also
1	Double-click on the element in a diagram or in the Project Browser The element Properties dialog displays	
2	Click on the Requirements tab	
3	Locate and highlight the requirement	
4	Click on the Move External button The Browse Project dialog displays	
5	Select the package to place the new requirement in	

Step	Action	See Also
6	Click on the OK button	

Enterprise Architect creates a new Requirement element in the target package and automatically generates a Realization connector from the element to the Requirement.

The requirement is now marked external and the dialog fields grayed out. To edit its details, double-click on the requirement.

The requirement element is now located in the target package.

Learn More:

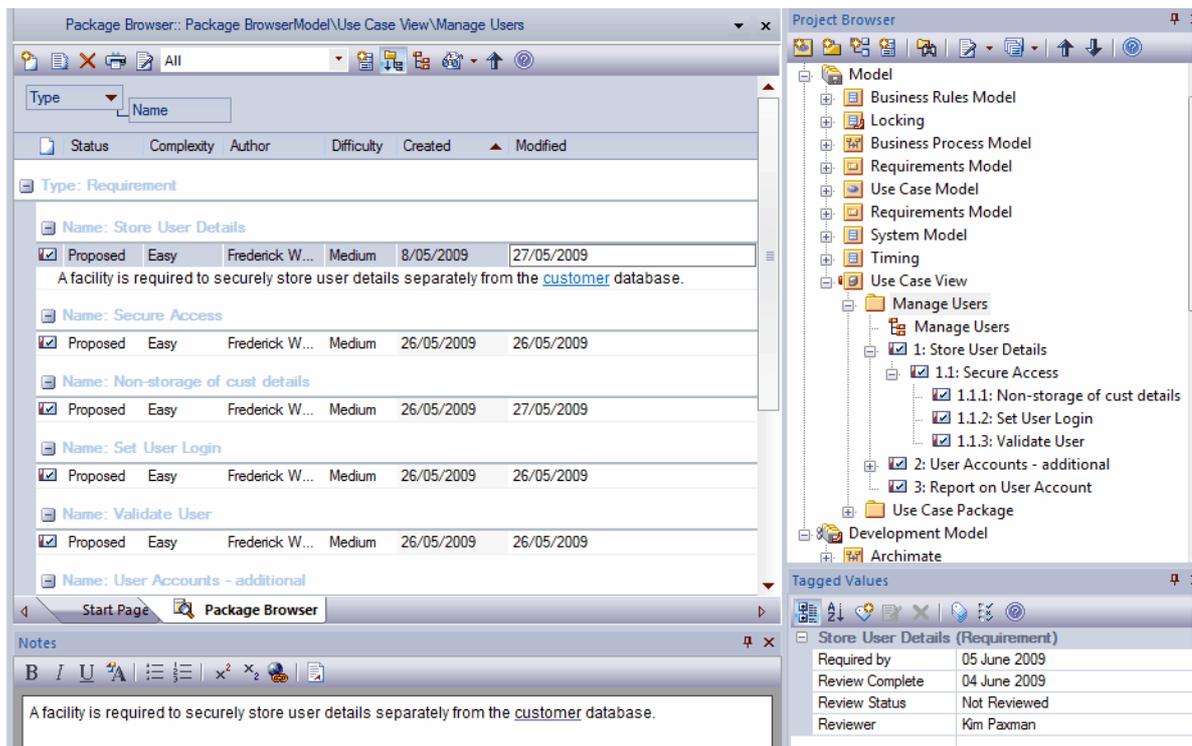
- [Requirement Properties](#) 1157

7.4 Manage Requirements

One of the main advantages of managing Requirements in Enterprise Architect is that you can display many different aspects of information on the Requirements, from overall organization and location through lists, current status, general properties, detailed individual notes and specific properties, to relationships and dependencies with other elements in the model.

As most of these aspects are displayed in dockable windows, you can review the Requirements from several different perspectives simultaneously in the Enterprise Architect work area, as shown below:

Example:



This display shows the position of the *Store User Details* Requirement element in the model, and how it relates to other Requirements (**Project Browser**); the default characteristics of the Requirement (**Package Browser**) and the extended characteristics (**Tagged Values** window), and a detailed description of the Requirement (**Notes** window). You can configure some of these windows to display more information, and/or use other windows and facilities.

Learn More:

- [View Requirements](#) ^[1170]
- [Trace Use of Requirements](#) ^[1170]
- [Manage Requirement Changes](#) ^[1171]
- [Report on Requirements](#) ^[1173]

7.4.1 View Requirements

Use the following windows and facilities to: locate and list Requirement elements in the model; add, move and delete the elements; display and edit the properties and characteristics of individual elements; and generate reports on packages or specific elements.

Topics:

Topic	Detail	See also
Usage	<p>Use the following windows and facilities to: locate and list Requirement elements in the model; add, move and delete the elements; display and edit the properties and characteristics of individual elements; and generate reports on packages or specific elements.</p> <ul style="list-style-type: none"> • Project Browser - shows the content and structure of your model • Diagram List - lists the elements in a diagram, filtered and sorted according to the settings you define; shows all or selected default properties of each element • Package Browser - lists the elements in a package, filtered and sorted according to the settings you define; shows all or selected default properties of each element • (Requirements) Diagram - shows the arrangement of a group of Requirements, and can show whether the elements are in the same package or different packages • Model Search - enables you to locate Requirements in general in the model, or specific Requirement elements, according to the search criteria you use • Model Views - enables you to maintain links to commonly-used elements, and to rapidly show developments and changes in (Requirement) package contents through either reports or slide shows of selected diagrams • Properties - shows every standard property of a selected element, whether updated by the user or maintained automatically by the system • Tagged Values - shows extended properties of a selected Requirement element • Element Browser - shows every added-on property, such as attributes, operations, Tagged Values and constraints • Notes - displays the detailed description of a requirement, and any other additional information recorded on the requirement 	<p>Project Browser ^[44]</p> <p>Diagram List ^[462]</p> <p>Package Browser ^[456]</p> <p>Requirements Diagram ^[1174] ^[540]</p> <p>Model Search ^[475]</p> <p>Model Views ^[464]</p> <p>Properties ^[689]</p> <p>Tagged Values ^[762]</p> <p>Element Browser ^[687]</p> <p>Notes ^[769]</p>

7.4.2 Trace Use of Requirements

Topics:

Topic	Detail	See also
Usage	<p>Having investigated the representation of requirements in your model, you might review either how they have been used to direct development through the model, or how a particular development was initiated. This is discussed in greater detail in the Traceability topics, but the windows and facilities you might use to follow development</p>	<p>Traceability ^[493]</p> <p>Connect Requirements ^[1167]</p> <p>Relationship</p>

Topic	Detail	See also
	<p>from Requirements are briefly described below. The significant feature in tracing Requirements and development is the connectors between the elements.</p> <ul style="list-style-type: none"> • Relationships window - quickly identifies every relationship of which a selected Requirement element is a member, and the partner element in that relationship, whether or not the relationship is visible in the current diagram. If the partner element is not in the diagram, you have the option of adding it • Traceability window - a very useful tool in showing chains of relationships that include the selected element. The window can show, for example, that a Requirement A is realized by a Use Case X, but Use Case X also realizes Requirement B, which in turn is also realized by Use Case Y. You can control the type and extent of these relationship chains, but as the system checks the connectors and partner elements of every relationship within the limits you impose, the system can take some time to produce the final results • Relationship Matrix - a significant tool in mapping the relationships between the Requirements elements in a package and other elements in either that package or a different package. Where a relationship is missing, you can add it; if an existing relationship is misplaced, you can delete it • Properties dialog, Requirements tab - for elements other than Requirements (particularly Use Cases), shows all internal responsibilities and external requirement elements attached to the element • Scenarios & Requirements window - as for the Properties dialog, shows the Requirements and responsibilities of the selected element, and the scenarios and constraints under which the Requirements are being realized 	<p>Window^[504]</p> <p>Traceability window^[495]</p> <p>Relationship Matrix^[496]</p> <p>Properties Dialog^[660]</p> <p>Requirements^[664]</p> <p>Scenarios & Requirements^[689]</p>

7.4.3 Manage Requirement Changes

Because requirements are statements of what a system or process must do or provide, they have a great impact on the modeling and development of the system. A new requirement might initiate an extensive program of work, and changes to or removal of that requirement can therefore have a major effect on the model. Issues concerning requirements, and changes to Requirements, must both be carefully managed.

The first steps in managing changes to requirements would be to raise *specific Issue and Change* request items against the Requirement element. You could monitor the appearance of these items using the filtered searches of the *Model Views*. You might then **review** the Requirement properties and/or its relationship hierarchies. During model development, you might capture periodic *Baselines* and use these to review the changes and, if necessary, roll them back to a previous point. You might also use the *Auditing* facility to monitor changes as they are made, and to ensure that no unauthorized or potentially risky changes are being made in the model.

These facilities are briefly discussed below.

Topics:

Topic	Detail	See also
Changes and Issues	A change is, very broadly, an item defining an addition or alteration to a requirement. An issue identifies either a failure to meet a requirement,	<p>Changes^[1721]</p> <p>Issues^[1719]</p>

Topic	Detail	See also
	<p>or a risk in meeting the requirement.</p> <p>Changes and issues can arise in development at a number of levels, being raised for problems that apply system-wide down to within a specific element. There are two mechanisms that can be used to identify a change or issue, and the work required to resolve it:</p> <ul style="list-style-type: none"> • Change and Issue (or Defect) elements - structured comments that identify a problem at system-level, although they can also be attached to a specific element from which a problem arises. Both types of element resemble the Requirement element, and can be linked to one or more other elements that have to be reviewed, with relationships such as Association, Dependency and Realize. The two types of element can also form hierarchies or groups, where complex problems arise • Maintenance items raised against a specific element, and recorded for that element in the Maintenance window. Maintenance items enable the distinction between Defects (a failure to meet a requirement) and Issues (a risk factor that might affect satisfying the requirement). They also include Tasks, which record work items associated with the element <p>Maintenance items are very specific, but if there is a possibility of an item having a wider impact on other elements or the system in general, you can translate the item into a Change or Issue element, or any other type of element that best identifies the problem and its solution.</p>	<p>Maintenance items ^[1713]</p> <p>Maintenance Workspace ^[1713]</p> <p>Create an Element From a Maintenance Item ^[1717]</p>
Model Views	<p>Model Views are very useful for trapping changes and issues in the model, especially on Requirements. You can set up searches to identify the appearance of new Change or Issue elements, or to detect changes in the properties of the Requirement elements themselves.</p>	<p>Model Views ^[464]</p>
Baselines	<p>A Baseline is a snapshot of a package or a model branch at a particular point in time, which you determine. You can use the Baseline as a distribution mechanism for changes to the model, but the main use is to enable you to compare the current model with a previous stage, and detect what changes have been made since the Baseline was captured. If you do not want a change to remain in the model, you can roll the affected elements back to the state they had in the Baseline. Therefore, if you maintain your requirements in a specific package or branch, you can capture Baselines of the package and ensure that changes conform to your change management process or, if not, can be reversed.</p>	<p>Baselines ^[370]</p>
Auditing	<p>The Auditing facility enables you to capture any changes made to your model within the selection criteria that you define. You can, for example, configure the Auditing facility to specifically record changes to Requirement elements. As auditing is continuously monitoring, you can detect changes as they are made, and verify that they are acceptable. You can also store the log of changes, and review it later on. Note that you cannot reverse the changes automatically, as you can with Baselines. You might therefore use Auditing to identify changes to investigate more fully and - if necessary - reverse in a Baseline comparison.</p>	<p>Auditing ^[298]</p> <p>Auditing Settings ^[300]</p>

Learn More:

- [View Requirements](#) ^[1170]

7.4.4 Report on Requirements

Topics:

Topic	Detail	See also
Usage	<p>Enterprise Architect provides two report generation facilities that enable you to output RTF reports and HTML reports on your model structure and components. The RTF reporting facility is especially comprehensive, and contains a number of features that provide particular support to Requirements Management:</p> <ul style="list-style-type: none"> • A requirements report template that extracts details of external requirements in the model; you can copy and tailor this template for your particular requirements • Options in the Diagram List, Package Browser and Model Search to generate RTF reports on selected (Requirement) items from the collected information • The Implementation Report, which lists for a selected package the elements that require implementers, together with any source elements in Realize (Implements) relationships with those elements • The Dependency Report, which lists for a selected package any elements that are dependent on another element for their specification; for example, a Use Case derives its specification from the Requirement that it realizes 	<p>RTF Document ^[1726]</p> <p>RTF Dialog Options ^[1733]</p> <p>Diagram List ^[462]</p> <p>Package Browser ^[456]</p> <p>Model Search ^[475]</p> <p>Implementation Report ^[1785]</p> <p>Realization ^[1007]</p> <p>Dependency Report ^[1783]</p>

7.5 Requirements Diagram

A Requirements diagram is a custom diagram used to describe a system's requirements or features as a visual model.

Requirements are defined using Requirement elements (Custom elements of type Requirement). To view the detailed description of a Requirement, double-click on the element to display its properties.

Requirements can have relationships with other elements such as other Requirements and Use Cases. To view the traceability of a requirement, use the Traceability window.

Use To:

- Describe a system's requirements or features as a visual model
- Link Requirement elements to Use Cases and Components in the system to illustrate how a particular system requirement is met
- Enable traceability between specifications and design requirements, and the model elements that realize them

Example Diagram: [Example Requirements Diagram](#)^[1174]

Tools:

Select Requirements diagram elements and connectors from the Requirements pages of the Toolbox.

Requirements Diagram Elements	Requirements Diagram Connectors
 Package	 Aggregate
 Requirement	 Inheritance
 Feature	 Associate
 Object	 Implements

Learn More:

- [Trace - Tracking Dependencies](#)^[493]
- [The Traceability Window](#)^[495]

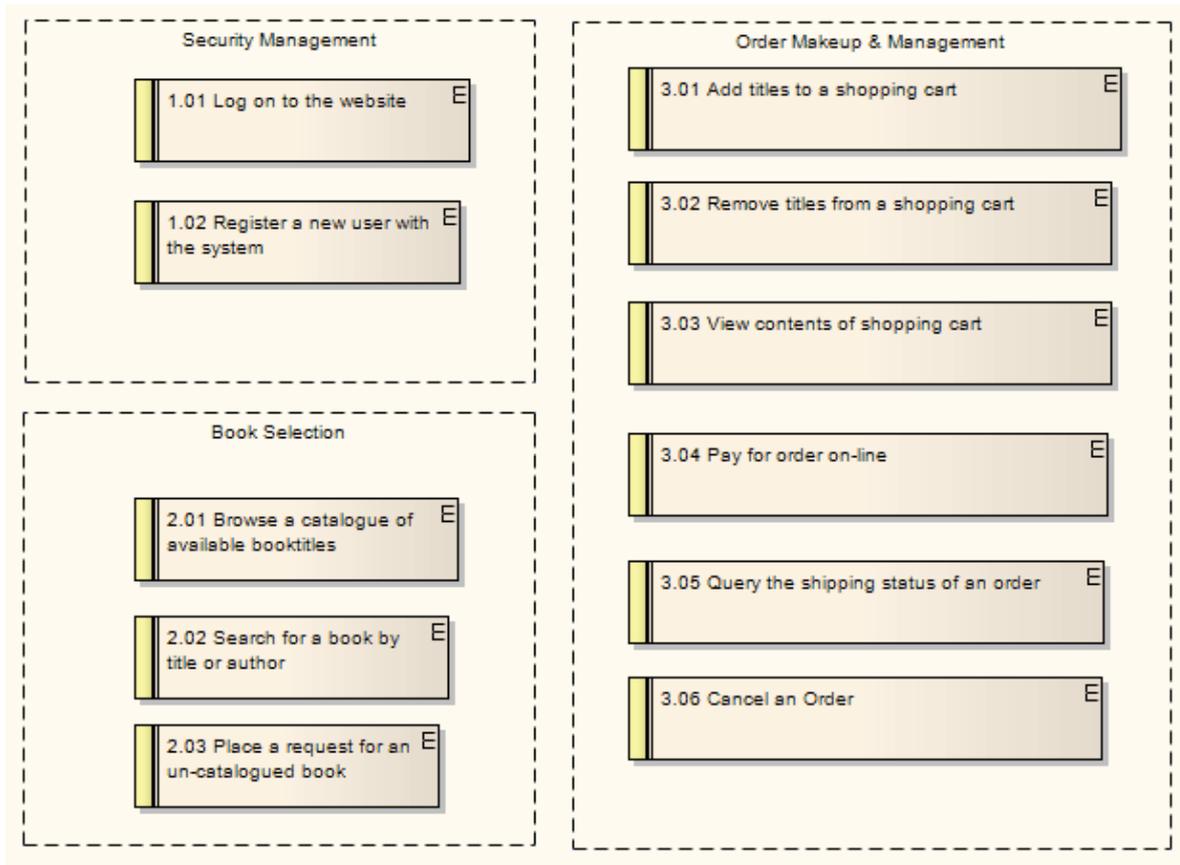
7.5.1 Example Requirements Diagram

The example below illustrates a Requirements diagram.

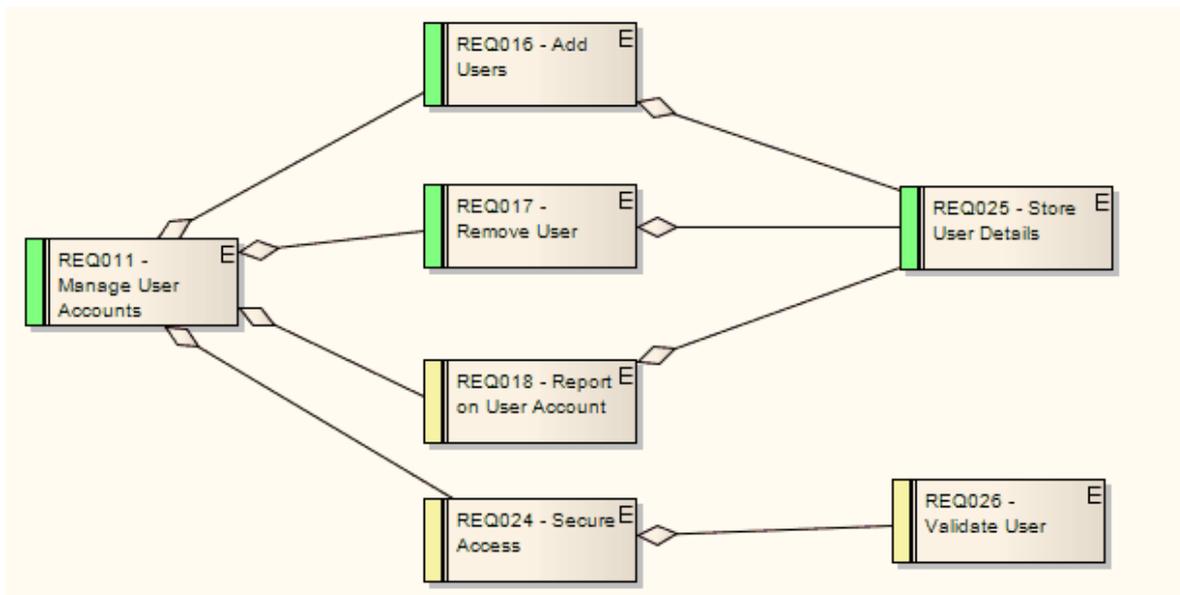
Requirement elements can be linked back to Use Cases and Components in the system to illustrate how a particular system requirement is met.

Change and Defect (Issue) elements look the same as Requirement elements and can be coded and managed in the same way.

Example 1:



Example 2:

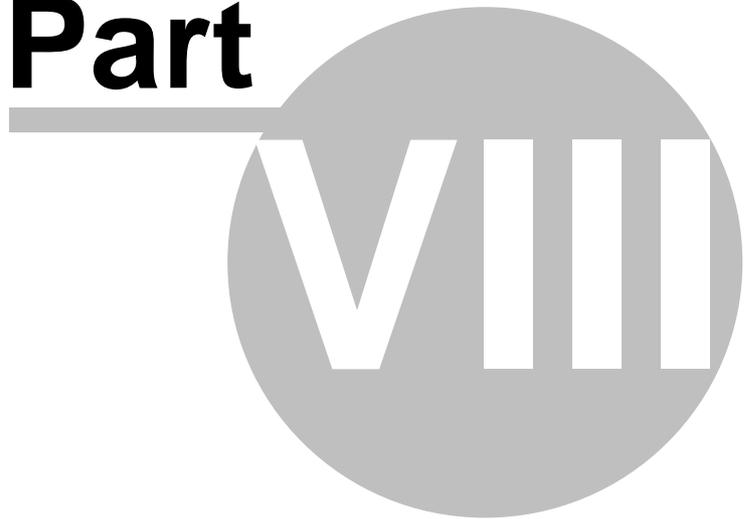


Learn More:

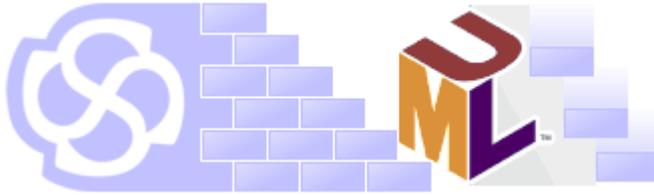
- [Requirements Diagram](#) [1174]
- [Requirements](#) [1154]

- [Use Case](#)^[935]
- [Component](#)^[948]
- [Changes](#)^[1721]
- [Defects \(Issues\)](#)^[1719]

Part



8 Domain Based Models



Enterprise Architect provides specific modeling tools for a range of specialized model types, as outlined below.

Enterprise Architect also provides an additional set of extended diagrams and a set of inbuilt and extension stereotype elements for more general use, and supports you in developing your own modeling languages.

Topic	Detail	See also
Requirements	Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model	Requirements modeling ^[1152]
Business Modeling	Modeling the business process is an essential part of any software development process, enabling the analyst to capture the broad outline and procedures that govern what it is a business does	Business modeling ^[1193]
Business Rules	Business Rule modeling captures the rules that govern a business, and their relationships with the entities and specific tasks within the organization or system	Business Rule modeling ^[1199]
BPMN	The Business Process Modeling Notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles; these profiles enable you to develop BPMN diagrams quickly and simply	BPMN modeling ^[1219]
BPEL	Business Process Execution Language is an executable language for specifying interactions with Web Services Enterprise Architect uses the BPMN profile as a graphical front-end to capture BPEL Process descriptions	BPEL modeling ^[1235]
SysML	SysML is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that might include hardware, software, information, personnel, procedures and facilities	Systems engineering (SysML) ^[1548]
Data Modeling	Enterprise Architect provides easy-to-use tools for building and maintaining all of the fundamental data models - Conceptual, Logical and Physical; because Enterprise Architect lets you visualize each type of data model in the same repository, you can easily manage dependencies between each level of abstraction	Data modeling ^[1276]
XSD	Enterprise Architect supports rapid modeling, forward	XML Schema modeling

Topic	Detail	See also
	engineering and reverse engineering of W3C XML schemas (XSD), critical for the development of a complete Service Oriented Architecture (SOA)	(XSD) ^[158]
WSDL	Enterprise Architect enables rapid modeling, forward engineering and reverse engineering of W3C XML Web Service Definition Language (WSDL), critical for the development of a complete Service Oriented Architecture (SOA)	Web Service modeling (WSDL) ^[161]
SPEM	The Software and Systems Process Engineering Meta-model (SPEM) is a conceptual framework for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM	Software Process modeling (SPEM) ^[406]
ArchiMate	ArchiMate is an open-standard enterprise architecture language based on the IEEE 1471 standard, providing a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure It enables Enterprise Architects to clearly describe, analyse and visualize the relationships among business domains	ArchiMate ^[272]
Data Flow Diagrams	A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, and can also be used to visualize data processing (structured design) Developing a DFD helps in identifying the transaction data in the data model	Data Flow Diagrams ^[184]
Entity Relationship Diagrams	Entity-relationship modeling is an abstract and conceptual database modeling method, used to produce a schema or semantic data model of, for example, a relational database and its requirements, visualized in Entity-Relationship Diagrams (ERDs) ERDs in Enterprise Architect assist you in building conceptual data models through to generating Data Definition Language (DDL) for the target DBMS	Entity Relationship Diagrams ^[382]
Eriksson-Penker Extensions	Eriksson-Penker extensions provide a framework for UML business processing model extensions, to which an Enterprise Architect can add stereotypes and properties appropriate to their business In Enterprise Architect, the Eriksson-Penker profile provides, through a set of stereotypes, a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization	Eriksson-Penker Extensions ^[273]
Group of Four Patterns	Gang of Four (GoF) Patterns are 23 classic software design patterns providing recurring solutions to	GoF Patterns ^[545]

Topic	Detail	See also
	<p>common problems in software design</p> <p>Enterprise Architect provides each pattern through an icon in the Diagram Toolbox</p>	
ICONIX	<p>The ICONIX Process is a streamlined approach to Use Case driven UML modeling that uses a core subset of UML diagrams and techniques to provide thorough coverage of object-oriented analysis and design</p> <p>Its main activity is robustness analysis, a method for bridging the gap between analysis and design</p>	ICONIX ^[1542]
Mind Mapping	<p>A Mind Map is an image-centered diagram used to represent semantic or other connections between words, ideas, tasks or other items arranged radially around a central key word or idea</p> <p>A Mind Map is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing</p>	Mind Mapping ^[1182]
SoaML	<p>Service Oriented Architecture (SOA) is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results</p>	SoaML ^[1624]
SOMF	<p>The service-oriented modeling framework (SOMF) is a service-oriented development life cycle methodology, offering a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle management and modeling</p>	SOMF ^[1628]
Extended Diagrams	<p>Enterprise Architect provides an additional set of diagram types that extend the core UML diagrams for domain-specific models</p> <p>Also, the specialized modeling tools listed in the first part of this table each have their own specialized diagrams</p>	<p>Analysis Diagram^[1187]</p> <p>Custom Diagram^[1185]</p> <p>Requirements Diagram^[1174]</p> <p>Maintenance Diagram^[1711]</p> <p>User Interface Diagram^[1287]</p> <p>Database Diagram^[1278]</p> <p>Business Modeling and Business Interaction Diagrams^[1191]</p>
Inbuilt and Extension Stereotypes	<p>Behavioral and Structural elements can be extended through the use of stereotypes; Enterprise Architect provides a number of inbuilt extensions</p>	Inbuilt and Extension Stereotypes ^[1288]
Build Your Own Modeling Language	<p>Enterprise Architect enables you to extend the scope both of your modeling and of the UML components you use, through the use of stereotypes, profiles and patterns to develop your own modeling applications</p>	Build Your Own Modeling Language ^[1038]

8.1 Domain Based Diagrams

In addition to diagrams defined by the UML, Enterprise Architect provides some extended diagram platforms to model business processes or develop custom diagrams.

Diagram Type	Detail	See also
Analysis Diagram	An Analysis diagram is a simplified Activity diagram, which is used to capture high level business processes and early models of system behavior and elements	Analysis Diagram ^[1187]
Custom Diagram	A Custom diagram is an extended Class diagram that is used to capture requirements, user interfaces or custom-design models	Custom Diagram ^[1183]
Requirements Diagram	A Requirements diagram is a custom diagram used to describe a system's requirements or features as a visual model	Requirements Diagram ^[1174]
Maintenance Diagram	A Maintenance diagram is a custom diagram used to describe change requests and issue items within a system model	Maintenance Diagram ^[1711]
User Interface Diagram	User Interface diagrams are custom diagrams used to visually mock-up a system's user interface using forms, controls and labels	User Interface Diagram ^[1281]
Data Modeling Diagram	A Data Modeling diagram is a Class diagram used for representing database schemas	Data Modeling Diagram ^[1278]
Documentation	Virtual documents enable you to structure and filter your RTF and HTML reports by selecting, grouping and ordering individual packages independent of the organization of the Project Browser	Documentation ^[1774]
Business Modeling and Business Interaction	Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior of a business system Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram	Business Modeling and Business Interaction ^[1191]

8.2 Analysis Models

See:

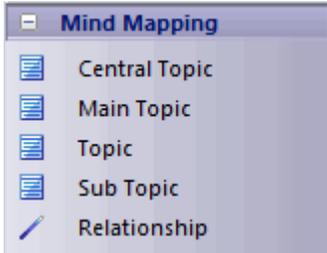
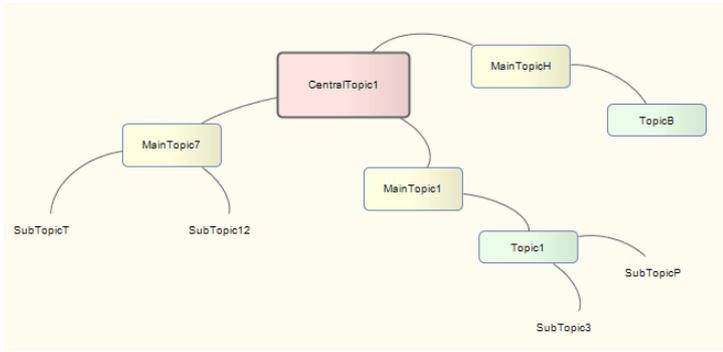
- [Mind Mapping](#)^[1182]
- [Custom Diagram](#)^[1183]
- [Process](#)^[1297]
- [Event](#)^[1292]
- [Data Flow Diagrams](#)^[1184]
- [Analysis Stereotypes](#)^[1186]
- [Control](#)^[1289]
- [Analysis Diagram](#)^[1187]
- [Boundary](#)^[1288]
- [Entity](#)^[1290]

8.2.1 Mind Mapping

Access: From the Toolbox select: **More Tools | Mind Mapping**

Topics:

Topic	Detail	See also
Introduction	<p>The following text is derived from the Mind Map entry in the online Wikipedia.</p> <p><i>A Mind Map is a diagram used to represent words, ideas, tasks or other items linked to and arranged radially around a central key word or idea. It is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.</i></p> <p><i>A Mind Map is an image-centered diagram that represents semantic or other connections between portions of information. By presenting these connections in a radial, non-linear graphical manner, it encourages a brainstorming approach to any given organizational task, eliminating the hurdle of initially establishing an intrinsically appropriate or relevant conceptual framework to work within.</i></p> <p><i>The elements are arranged intuitively according to the importance of the concepts and are organized into groupings, branches, or areas. The uniform graphic formulation of the semantic structure of information on the method of gathering knowledge, may aid recall of existing memories.</i></p> <p>The use of the term Mind Maps is trademarked in the UK and the USA by The Buzan Organization, Ltd.</p>	<p>Mind Map (Online Resource)</p>
Mind Mapping in Enterprise Architect	<p>Enterprise Architect enables you to develop Mind Maps quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The Mind Mapping facilities are provided in the form of:</p> <ul style="list-style-type: none"> • A Mind Mapping diagram type, accessed through the New Diagram dialog • A Mind Mapping page in the Toolbox • Mind Mapping element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	<p>New Diagram^[568]</p> <p>Toolbox Shortcut^[551]</p> <p>Quick Linker^[621]</p>

Topic	Detail	See also
<p>Mind Mapping Toolbox Page</p>	<p>The following icons are available:</p>  <ul style="list-style-type: none"> • <i>Central Topic</i> is the main theme of the Mind Map; you would normally have one or two of these on the diagram, but can add as many as are necessary • <i>Main Topic</i> represents the immediate concepts generated by the Central Topic • <i>Topic</i> represents the larger divisions of a Main Topic • <i>Sub Topic</i> represents the finer divisions of a Topic or Main Topic; you could also have Subtopics of Subtopics to represent increasingly finer distinctions • <i>Relationship</i> represents the connection between any two elements; you can have several Relationships per element. Each relationship has three anchor points, so you can curve the lines to develop the flow of concepts more easily <p>When dragged onto a Mind Mapping diagram, the elements and relationship have the following appearances:</p>  <p>As the elements can represent any concept, object or relationship, you can use the full range of element properties and features to expand on what the element represents, including adding Note elements. However, to preserve the simplicity and readability of the diagram itself, you cannot display the element compartments on the diagram.</p>	
<p>Disable Mind Mapping</p>	<p>If you prefer not to use Mind Mapping in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	<p>MDG Technologies 10331</p>

8.2.2 Custom Diagram

A Custom diagram is an extended Class diagram that is used to capture requirements, user interfaces or custom-design models.

Custom models provide a few extensions to the UML model and enable some exploratory and non-rigorous

experimentation with model elements and diagrams.

Tools:

Select Custom diagram elements and connectors from the Custom pages of the Toolbox.

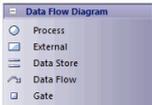
Custom Diagram Elements	Custom Diagram Connectors
 Package	 Associate
 Requirement	 Aggregate
 Issue	 Generalize
 Change	 Realize
 Screen	 Nesting
 UI Control	
 Test Case	
 Entity	

8.2.3 Data Flow Diagrams

Access: From the Toolbox select: **More Tools | Data Flow Diagrams**

Topics:

Images	Topic	Detail	See also
	Introduction	<p>The following text is derived from the Data Flow Diagram entry in the online Wikipedia.</p> <p><i>A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.</i></p> <p><i>Data flow diagrams were invented by Larry Constantine ... based on Martin and Estrin's "data flow graph" model of computation. (They) are one of the three essential perspectives of Structured Systems Analysis and Design Method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a dataflow diagram, users are able to visualize how</i></p>	<p>Data Flow Diagram (Online Resource)</p>

Images	Topic	Detail	See also
		<p><i>the system will operate, what the system will accomplish, and how the system will be implemented. The old system's dataflow diagrams can be drawn up and compared with the new system's dataflow diagrams to draw comparisons to implement a more efficient system.</i></p> <p><i>Developing a DFD helps in identifying the transaction data in the data model.</i></p>	
	<p>Data Flow Diagrams in Enterprise Architect</p>	<p>Enterprise Architect enables you to develop Data Flow diagrams quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The Data Flow diagram facilities are provided in the form of:</p> <ul style="list-style-type: none"> • A Data Flow diagram type, accessed through the New Diagram dialog • A Data Flow Diagram page in the Toolbox • Data Flow element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	<p>New Diagram [568]</p> <p>Toolbox Shortcut [55]</p> <p>Quick Linker [622]</p>
	<p>Data Flow Diagram Toolbox Page</p>	<p>The following icons are available:</p> <ul style="list-style-type: none"> • <i>Process</i> is a process or activity in which data is used or generated • <i>External</i> represents an external source, user or depository of the data • <i>Data Store</i> represents an internal physical or electronic repository of data, into and out of which data is stored and retrieved • <i>Data Flow</i> (connector) represents how data flows through the system, in physical or electronic form • <i>Gate</i> represents the termination point of incoming and outgoing messages on a lower level diagram (that is, messages to and from processes depicted elsewhere) <p>When dragged onto a Data Flow diagram, the elements and relationship have the following appearances:</p>	

Images	Topic	Detail	See also
		<p>To preserve the simplicity and readability of the diagram, you cannot display the element compartments on the diagram.</p>	
	Context Diagram	A <i>Context</i> diagram is a top-level Data Flow diagram that has just one Process element representing the system being modeled, showing its relationship to external systems.	
	Disable Data Flow Diagrams	If you prefer not to use Data Flow Diagramming in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	MDG Technologies <small>1033</small>

8.2.4 Analysis Stereotypes

Description

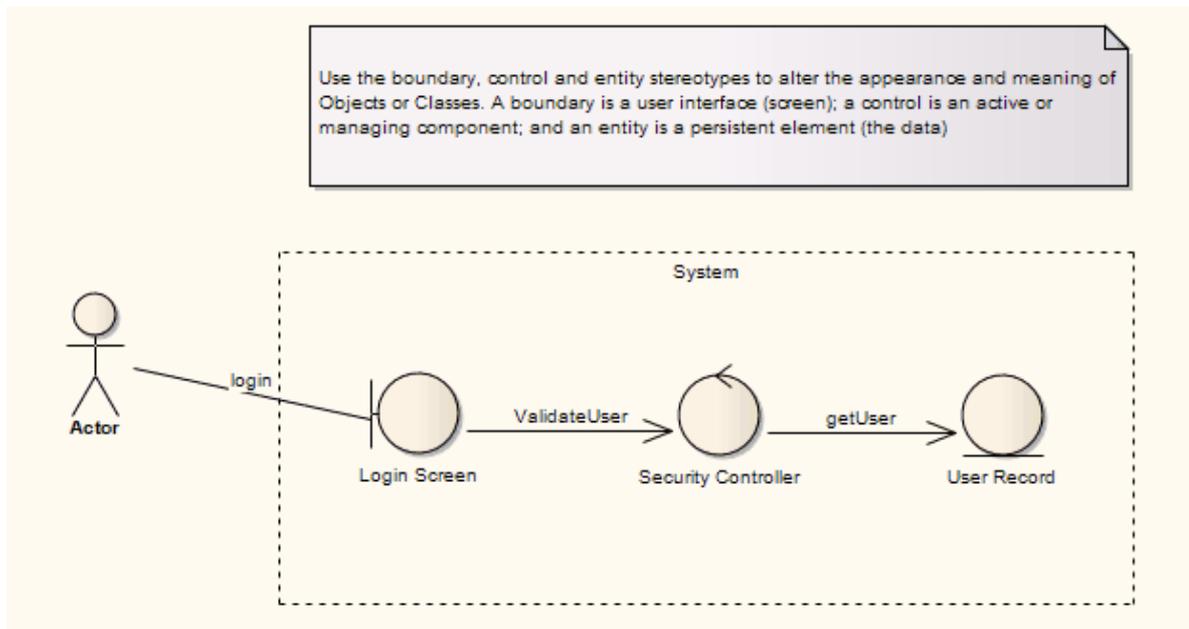
Enterprise Architect has some built in stereotypes that you can assign to an element during analysis. The effect of these stereotypes is to display a different icon from the normal element icon, providing a visual key to the element purpose. The *Robustness* diagram below illustrates the main types of inbuilt icons for elements.

The stereotypes used are:

- Boundary - for a system boundary (for example, a Login screen)
- Control - to specify an element is a controller of some process (as in the Model-View-Controller pattern)
- Entity - the element is a persistent or data element

Also see the Business Modeling elements, used in Business Modeling and Business Interaction diagrams.

Example



Use the boundary, control and entity stereotypes to alter the appearance and meaning of Objects or Classes. A boundary is a user interface (screen); a control is an active or managing component; and an entity is a persistent element (the data)

Learn More:

- [Boundary](#)^[1288]
- [Control](#)^[1289]
- [Entity](#)^[1290]
- [Business Modeling](#)^[1191]

8.2.5 Analysis Diagram

An Analysis diagram is a simplified Activity diagram, which is used to capture high level business processes and early models of system behavior and elements. It is less formal than some other diagrams, but provides a good means of capturing the essential business characteristics and requirements.

Robustness diagrams, used extensively in ICONIX, can be created as Analysis diagrams.

Example Diagram: [Example Analysis Diagram](#)^[1189]

Tools:

Select Analysis diagram elements and connectors from the Analysis pages of the Toolbox.

Analysis Diagram Elements	Analysis Diagram Connectors
Actor	Information Flow
Object	Object Flow
Process	Associate
Collaboration	Realize

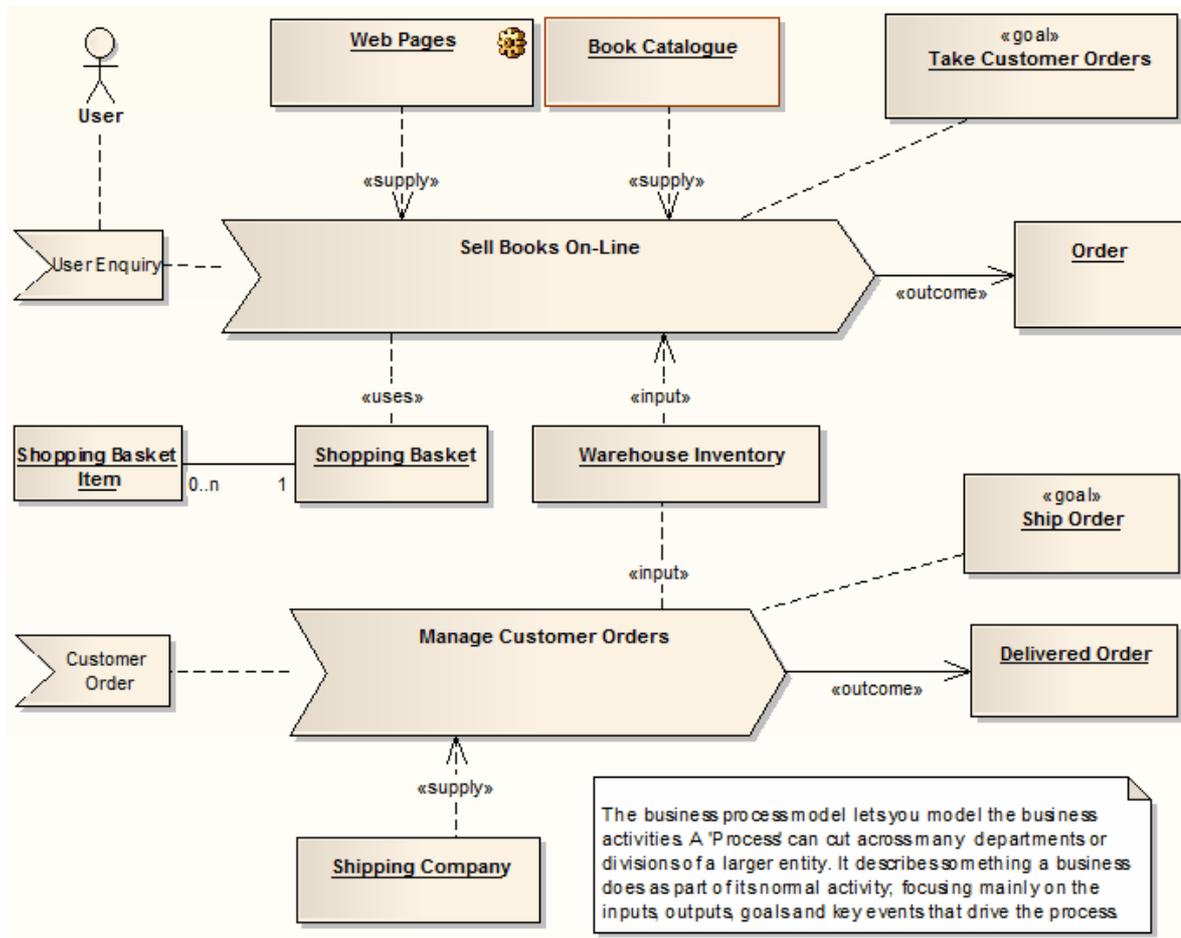
Analysis Diagram Elements	Analysis Diagram Connectors
 Collaboration Use	 Representation
 Send	
 Receive	
 Information	
 Information Item	
 Decision	
 Merge	
 Boundary	
 Control	
 Entity	

Learn More:

- [Activity Diagram](#) ^[81]
- [Business Process Modeling](#) ^[1193]
- [ICONIX](#) ^[1542]

8.2.5.1 Example Analysis Diagram

Enterprise Architect supports some of the Eriksson-Penker Business Extensions that facilitate business process modeling. The complete Eriksson-Penker Business Extensions UML Profile can also be loaded into Enterprise Architect and used to create detailed process models.



Learn More:

- [Eriksson-Penker Business Extensions](#) ¹²⁷³

8.3 Business Models

Enterprise Architect provides specific modeling tools for a range of analysis and Business Modeling types.

Topic	Detail	See also
Requirements	Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model	Requirements modeling ^[1152]
Business Modeling	Modeling the business process is an essential part of any software development process, enabling the analyst to capture the broad outline and procedures that govern what it is a business does	Business modeling ^[1193]
Business Rules	Business Rule modeling captures the rules that govern a business, and their relationships with the entities and specific tasks within the organization or system	Business Rule modeling ^[1199]
BPMN	The Business Process Modeling Notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles; these profiles enable you to develop BPMN diagrams quickly and simply	BPMN modeling ^[1219]
BPEL	Business Process Execution Language is an executable language for specifying interactions with Web Services Enterprise Architect uses the BPMN profile as a graphical front-end to capture BPEL Process descriptions	BPEL modeling ^[1235]
SPEM	The Software and Systems Process Engineering Meta-model (SPEM) is a conceptual framework for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM	Software Process modeling (SPEM) ^[408]
ArchiMate	ArchiMate is an open-standard enterprise architecture language based on the IEEE 1471 standard, providing a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure It enables Enterprise Architects to clearly describe, analyse and visualize the relationships among business domains	ArchiMate ^[1272]
Data Flow Diagrams	A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, and can also be used to visualize data processing (structured design) Developing a DFD helps in identifying the transaction data in the data model	Data Flow Diagrams ^[1184]

Topic	Detail	See also
Entity Relationship Diagrams	Entity-relationship modeling is an abstract and conceptual database modeling method, used to produce a schema or semantic data model of, for example, a relational database and its requirements, visualized in Entity-Relationship Diagrams (ERDs) ERDs in Enterprise Architect assist you in building conceptual data models through to generating Data Definition Language (DDL) for the target DBMS	Entity Relationship Diagrams ^[1382]
Eriksson-Penker Extensions	Eriksson-Penker extensions provide a framework for UML business processing model extensions, to which an Enterprise Architect can add stereotypes and properties appropriate to their business In Enterprise Architect, the Eriksson-Penker profile provides, through a set of stereotypes, a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization	Eriksson-Penker Extensions ^[1273]
Mind Mapping	A Mind Map is an image-centered diagram used to represent semantic or other connections between words, ideas, tasks or other items arranged radially around a central key word or idea A Mind Map is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing	Mind Mapping ^[1182]
SoaML	Service Oriented Architecture (SOA) is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results	SoaML ^[1624]
SOMF	The service-oriented modeling framework (SOMF) is a service-oriented development life cycle methodology, offering a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle management and modeling	SOMF ^[1628]

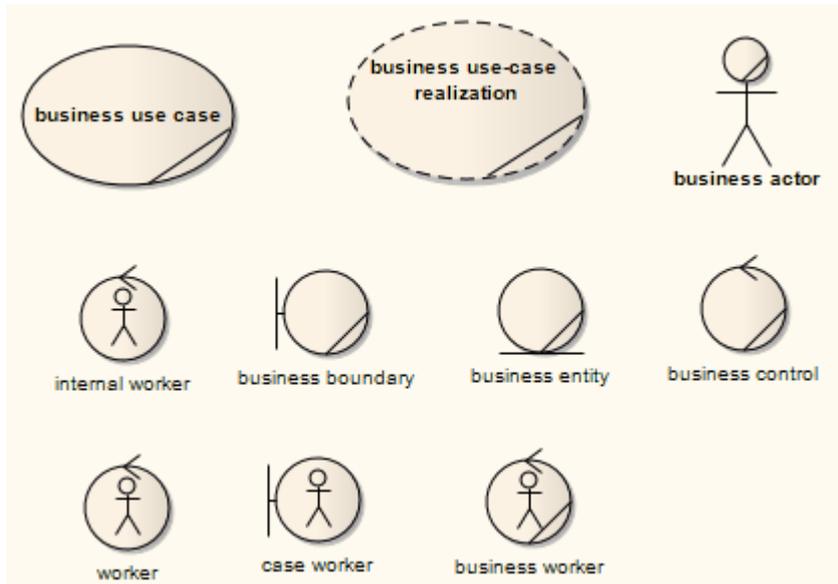
8.3.1 Business Modeling/Interaction

Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior of a business system. Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram. Both diagram types have the same default Toolbox, which consists of a Business Modeling element page. The available elements include stereotyped Objects, and a stereotyped Actor (Business Actor), Use Case (Business Use Case) and Collaboration (Business Use Case Realization).

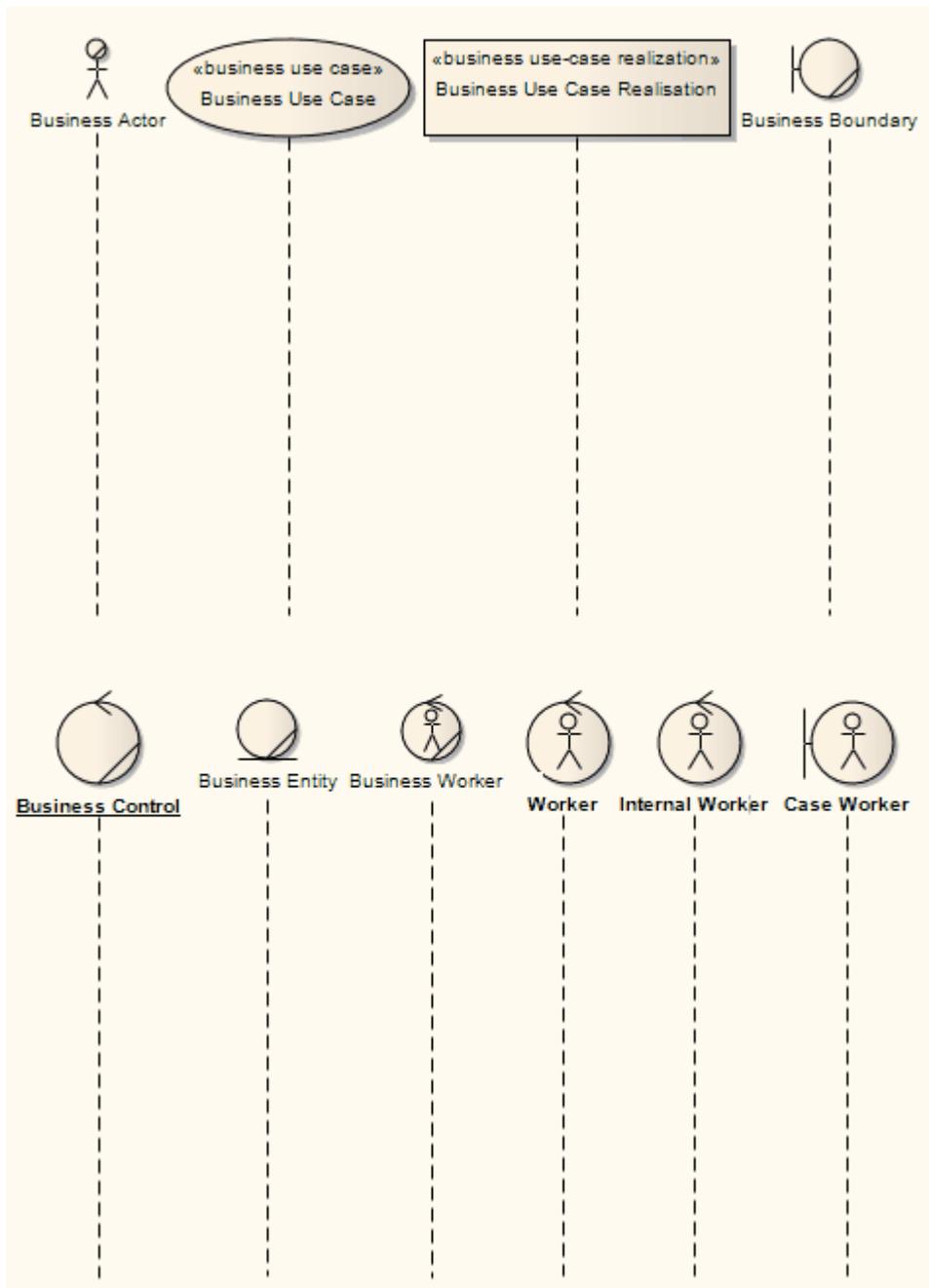
Example Diagram: [Example Business Modeling Diagram](#) ^[1192]

8.3.1.1 Example Business Modeling Diagram

The following diagram shows the appearance of the elements when dragged and dropped onto a Business Modeling diagram:



The following diagram shows the appearance of the elements when dragged and dropped onto a Business Interaction diagram:



8.3.2 Business Models

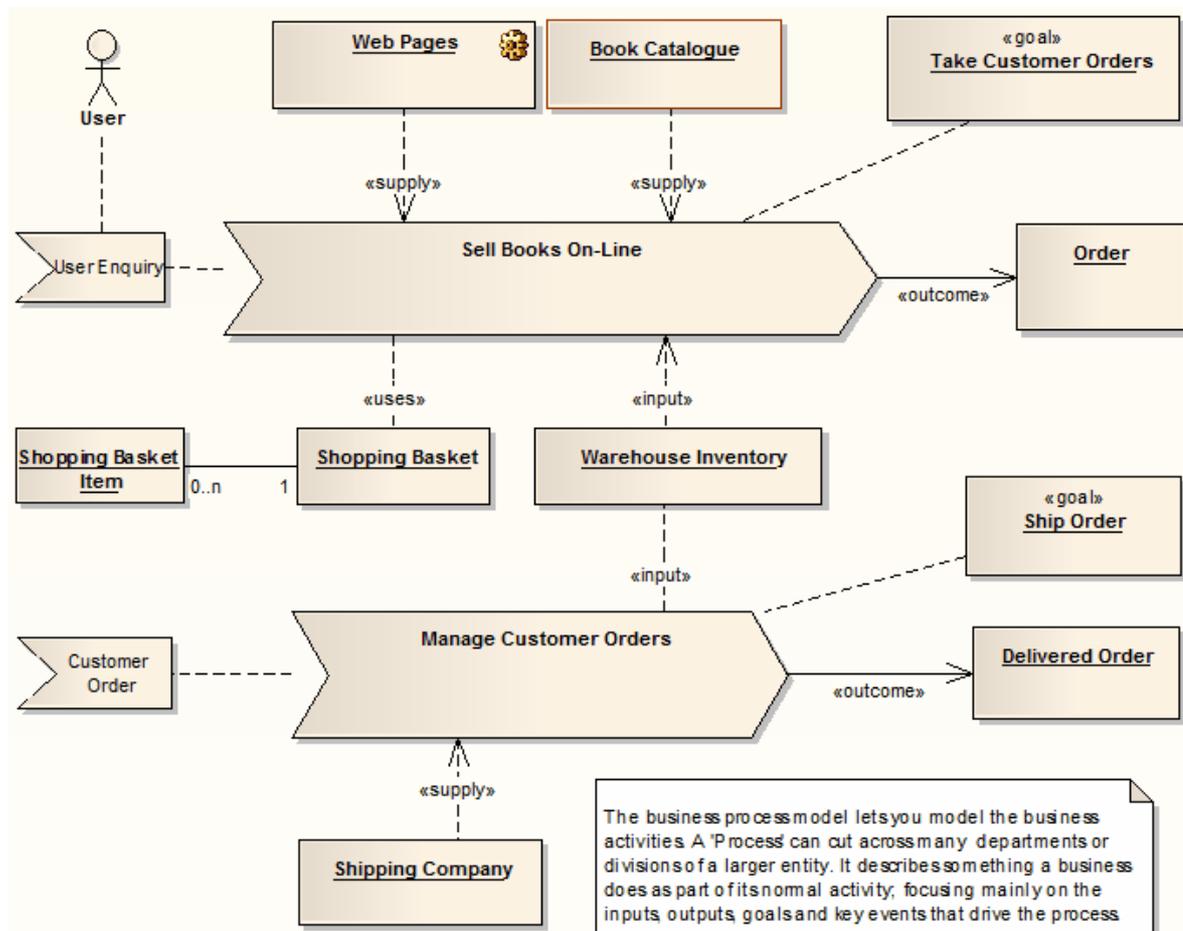
Topics:

Topic	Detail	See also
Modeling the Business Process	Modeling the business process is an essential part of any software development process. It enables the analyst to capture the broad outline and procedures that govern what it is a business does. This analysis model provides an overview of where the proposed software system being considered fits into the organizational structure and daily activities. It can also provide the justification for building the system by capturing the current manual and automated procedures	Analysis Model [1193]

Topic	Detail	See also
	<p>that are to be rolled up into a new system, and the associated cost benefit.</p> <p>As an early model of business activity, it enables the analyst to capture the significant events, inputs, resources and outputs associated with business process. By connecting later design elements (such as Use Cases) back to the business process model through Implementation connectors, it is possible to build up a fully traceable model from the broad process outlines to the functional requirements and eventually to the software artefacts actually being constructed.</p> <p>As the Business Process Model typically has a broader and more inclusive range than just the software system being considered, it also enables the analyst to clearly map what is in the scope of the proposed system and what is to be implemented in other ways (such as a manual process).</p>	

Example:

The example below demonstrates the kind of model that can be built up to represent a business process. In this model, the goal of the business process is to take customer orders and to ship those orders out. A user starts the process with an inquiry, which leads to the involvement of the Book Catalogue, Shopping Cart, on-line pages and warehouse inventory. The output of significance to the business is a customer order.



The second half of the process model is to respond to a customer order and ship the required items. The second process involves the warehouse inventory and shipping company, and completes when an order is delivered to the customer.

Learn More:

- [Business Modeling and Business Interaction Diagrams](#)^[1193]
- [Web Stereotypes](#)^[1284]

8.3.2.1 Analysis Models

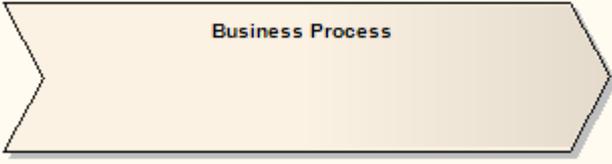
This section discusses the development of analysis models to construct business processes. It describes:

Topic	Link
Process Modeling Notation	Process Modeling Notation ^[1195]
Inputs, Resources and Information	Inputs, Resources and Information ^[1196]
Events	Events ^[1197]
Outputs	Outputs ^[1197]
Goals	Goals ^[1198]
A Complete Business Process	A Complete Business Process ^[1198]

8.3.2.1.1 Process Modeling Notation

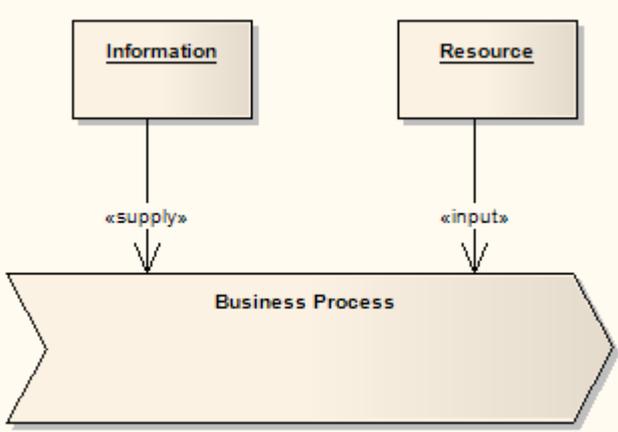
Topics:

Topic	Detail	See also
Abstract	<p>A business process model typically defines the following elements:</p> <ul style="list-style-type: none"> • The goal or reason for the process • Specific inputs • Specific outputs • Resources consumed • Activities that are performed in some order, and • Events that drive the process <p>The business process:</p> <ul style="list-style-type: none"> • Can affect more than one organizational unit • Can have a horizontal organizational impact • Creates value of some kind for the customer; customers can be internal or external <p>A business process is a collection of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how the work is done within an organization, in contrast to a product's focus on what. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs: a structure for action. The notation used to depict a business process is illustrated below.</p>	<p>Process^[1297]</p> <p>Events^[1197]</p> <p>Activity^[873]</p>

Topic	Detail	See also
	 <p>The process notation implies a flow of activities from left to right. Typically an Event element is placed to the left of the process and the output to the right. To specifically notate the internal activities, Activity elements can be placed inside the process element.</p>	
The BPMN File	<p>One popular notation and approach to business modeling is the Business Process Modeling Notation (BPMN). This notation is specifically targeted at the business modeling community and has a relatively direct mapping to UML through a BPMN Profile. Sparx Systems provides a built-in profile for BPMN modeling in Enterprise Architect.</p>	BPMN Models <small>[1219]</small>

8.3.2.1.2 Inputs, Resources and Information

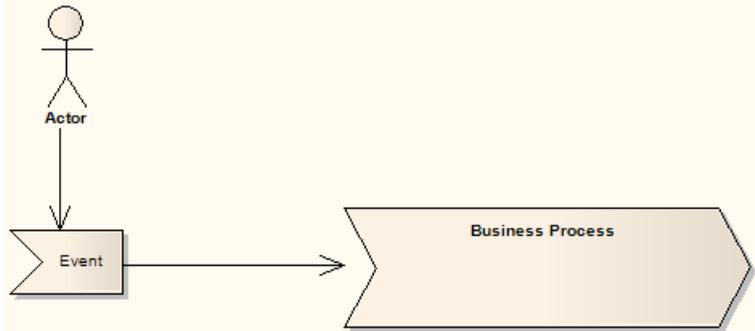
Topics:

Topic	Detail	See also
Abstract	<p>Business processes use information to tailor or complete their activities. Information, unlike resources, is not consumed in the process; rather it is used as part of the transformation process. Information can come from external sources, from customers, from internal organizational units and could even be the product of other processes. A resource is an input to a business process and, unlike information, is typically consumed during the processing. For example, as each daily train service is run and actuals recorded, the service resource is 'used up' as far as the process of recording actual train times is concerned.</p> <p>The notation to illustrate information and resources is shown below.</p>  <p>A <i>Supply</i> connector indicates that the information or object linked to the process is not used up in the processing phase. For example, order templates can be used over and over to provide new orders of a</p>	

Topic	Detail	See also
	<p>certain style; the templates are not altered or exhausted as part of this activity.</p> <p>An <i>Input</i> connector indicates that the attached object or resource is consumed in the processing procedure. As an example, as customer orders are processed they are completed and signed off, and typically are used only once per unique resource (order).</p>	

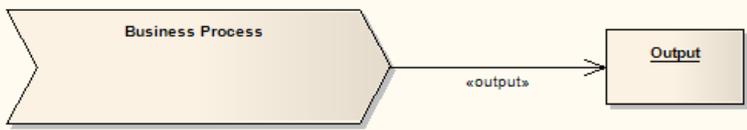
8.3.2.1.3 Events

Topics:

Topic	Detail	See also
Abstract	<p>An event is the receipt of some object, a time or date reached, a notification or some other trigger that initiates the business process. The event might be consumed and transformed (for example a customer order) or simply act as a catalyst (for example, nightly batch job).</p>  <p>The diagram illustrates the flow from an Actor to an Event and then to a Business Process. An Actor (represented by a stick figure) has a downward arrow pointing to an Event (represented by a trapezoidal shape with a pointed right side). From the Event, a horizontal arrow points to a Business Process (represented by a large arrow-shaped box pointing to the right).</p>	<p>Event^[1292]</p>

8.3.2.1.4 Outputs

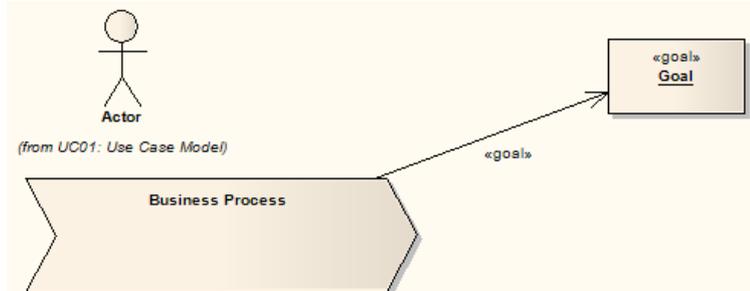
Topics:

Topic	Detail	See also
Abstract	<p>A business process typically produces one or more outputs of value to the business, either for internal use or to satisfy external requirements. An output might be a physical object (such as a report or invoice), a transformation of raw resources into a new arrangement (a daily schedule or roster) or an overall business result such as completing a customer order.</p> <p>An output of one business process might feed into another process, either as a requested item or a trigger to initiate new activities.</p>  <p>The diagram shows a Business Process (represented by a large arrow-shaped box pointing right) with an arrow labeled «output» pointing to an Output (represented by a rectangular box).</p> <p>An Output connector indicates that the business process produces</p>	

Topic	Detail	See also
	some object (either physical or logical) that is of value to the organization, either as an externally visible item or as an internal product (possibly feeding another process).	

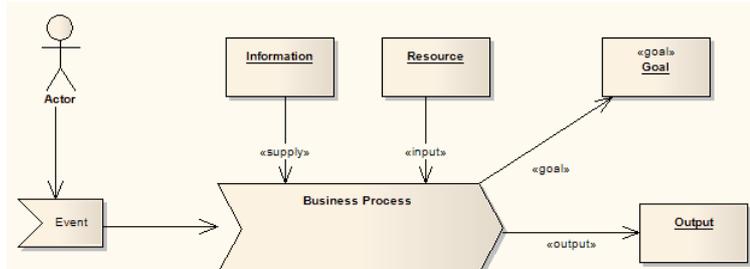
8.3.2.1.5 Goals

Topics:

Topic	Detail	See also
Abstract	<p>A business process has some well defined goal. This is the reason the organization does this work, and should be defined in terms of the benefits this process has for the organization as a whole and in satisfying the business requirements.</p>  <p>A Goal connector indicates that the attached object to the business process describes the goal of the process. A goal is the business justification for performing the activity.</p>	

8.3.2.1.6 A Complete Business Process

Topics:

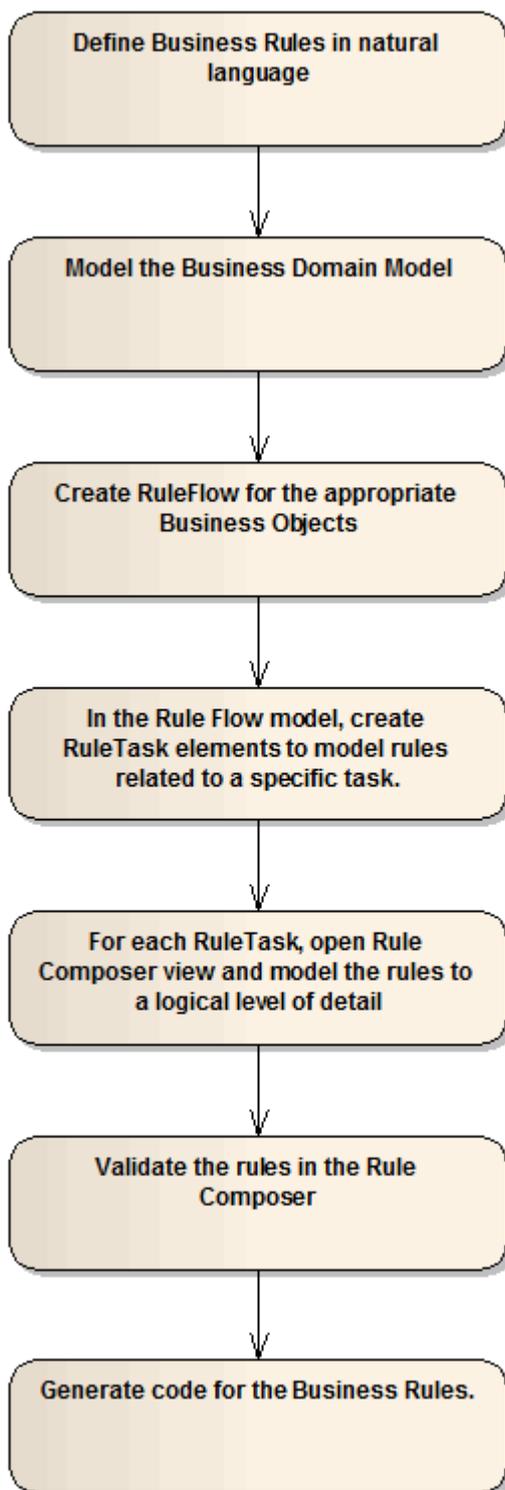
Topic	Detail	See also
Abstract	<p>The diagram below illustrates how the various model elements can be grouped together to produce a coherent picture of a named business process. Included are the inputs, outputs, events, goals and other resources that are of significance.</p> 	

8.3.3 *Business Rules*

To model Business Rules in Enterprise Architect, work through the following steps

Example:

These steps are represented graphically in the following flow:



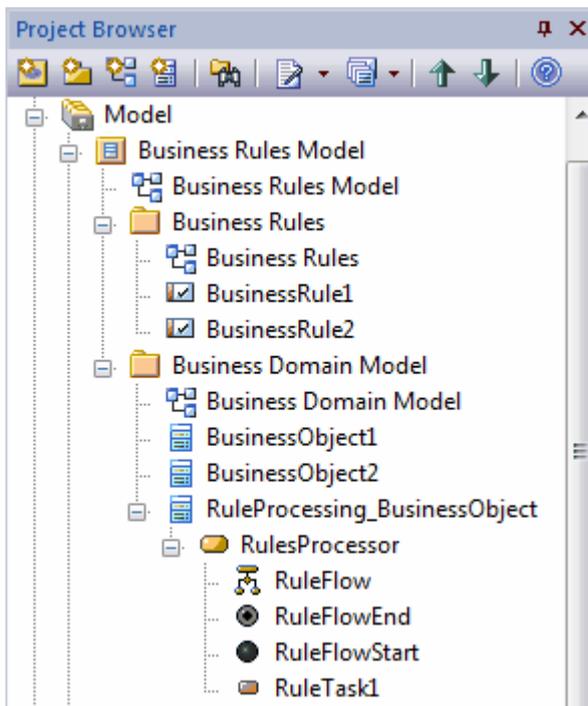
To create a Business Rule Model from a template provided with Enterprise Architect, follow the steps below

How To:

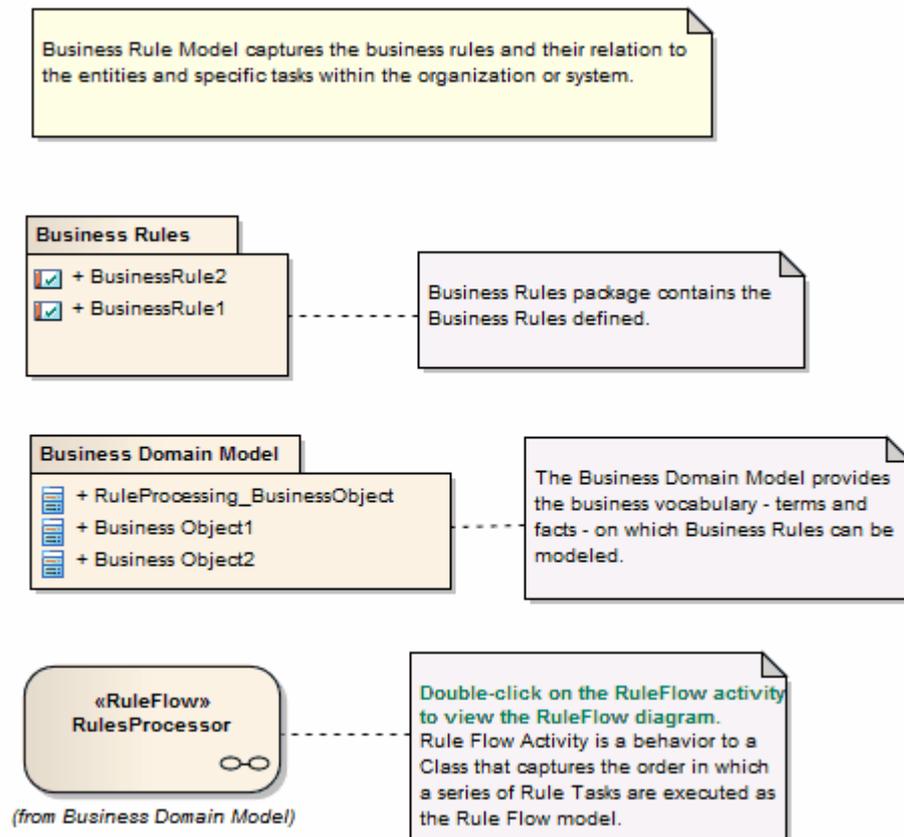
Step	Action	See Also
1	<p>In the Project Browser, either:</p> <ul style="list-style-type: none"> • Click on the New Model From Pattern icon in the toolbar • Right-click on a model root node and select the Add a New Model using Wizard context menu option • Right-click on a package and select the Add Add a New Model using Wizard context menu option <p>The Select Model(s) dialog displays.</p>	
2	<p>In the Select From field, click on the drop-down arrow and select Business Rule Model. Alternatively, if it is listed in the Technology panel, select the Business Rule Model item.</p>	
3	<p>In the Name panel, select the checkbox next to the Business Rule Model icon.</p>	
4	<p>Click on the OK button.</p>	

The following model structure is created in the **Project Browser**:

Example:



The *Business Rules Model* diagram, shown below, encapsulates the components of the Business Rules model.

**Notes:**

- Business Rule Modeling is available in the Business and Software Engineering edition and Ultimate edition of Enterprise Architect

Learn More:

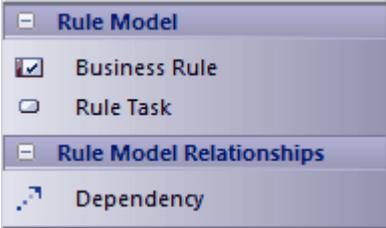
- [Create a Rule model](#) ^[1200]
- [Create a Business Domain Model](#) ^[1204]
- [Create a Rule Flow Model](#) ^[1206]
- [Compose Business Rules](#) ^[1211]
- [Validate Business Rules](#) ^[1216]
- [Code Generation for Business Rules](#) ^[1217]

8.3.3.1 Model Business Rules For RuleTasks

The *Rule Model* enables you to define *Business Rule* elements and associate them with a Rule Task. In the example, you might define a set of rules to perform an eligibility check for a customer, to determine if the customer is eligible to rent a car.

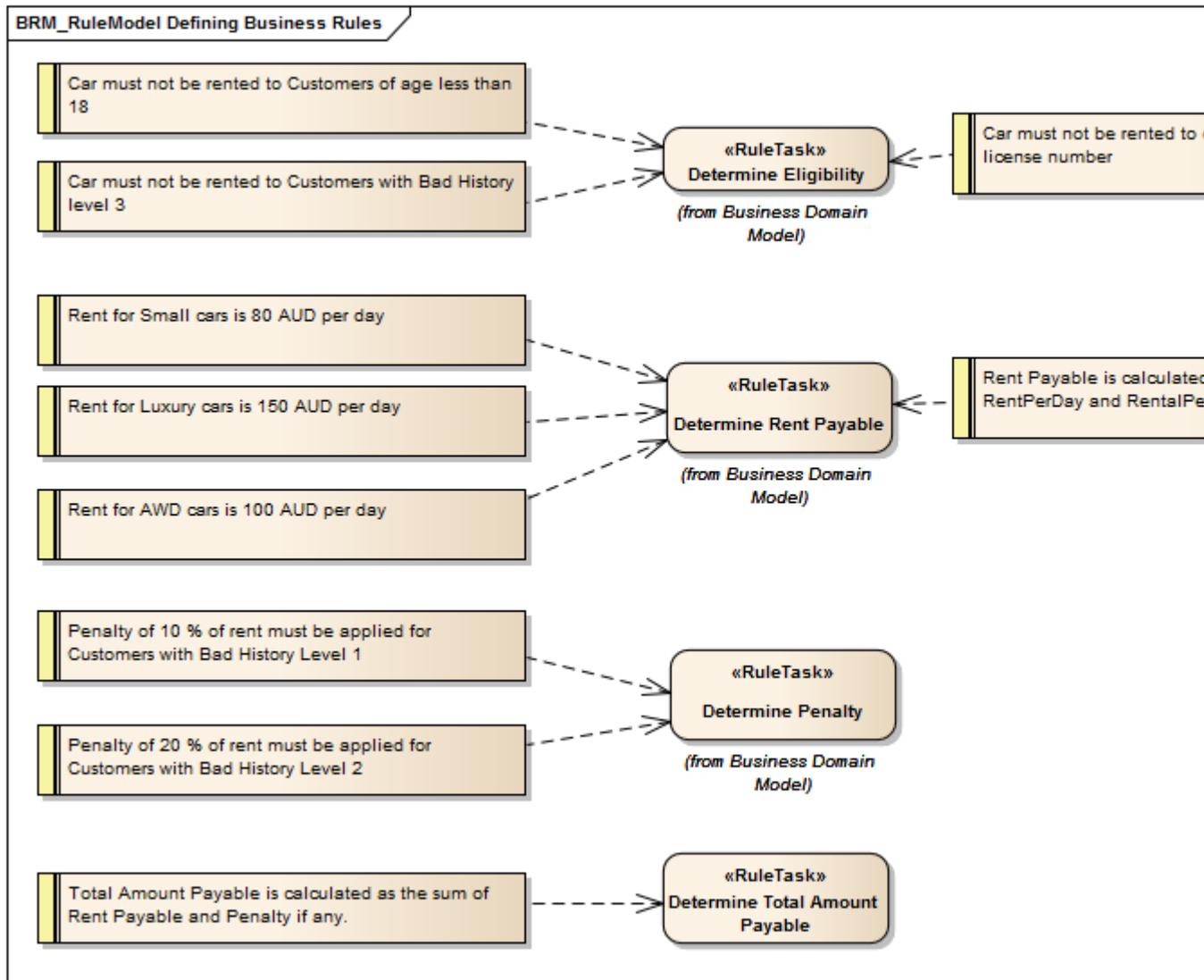
How To:

To define Business Rule elements and associate them with a Rule Task, follow the steps below:

Step	Action	See Also
1	Create a diagram of type <i>Rule Model</i> ; the Rule Model pages display in the Toolbox 	
2	Drag and drop a Rule Task element (<i>Determine Eligibility</i> in the example) from the Rule Flow Activity diagram package onto the Rule Model diagram	
3	Drag as many Business Rule elements as necessary from the Toolbox (or Project Browser if they exist already) onto the diagram You type the rule as the element name here, then define the parameters of the rules using the Rule Composer	Composing Business Rules <small>1217</small>
4	Create a Dependency relationship between each Business Rule element and the Rule Task element However, when you bring the rule into the Rule Composer, it automatically creates the Dependency relationship anyway	
5	Repeat steps 2 - 4 for the next Rule Task element	

Example:

The resulting Rule Model resembles the following diagram:



After you have modeled rules for all the Rule Task elements in the Rule Flow diagram, the Business Domain model is ready for **code transformation**. The code templates for generating technology-specific rule code work hand-in-hand with the **EASL code templates** to generate the code for the Rule Flow diagram.

Learn More:

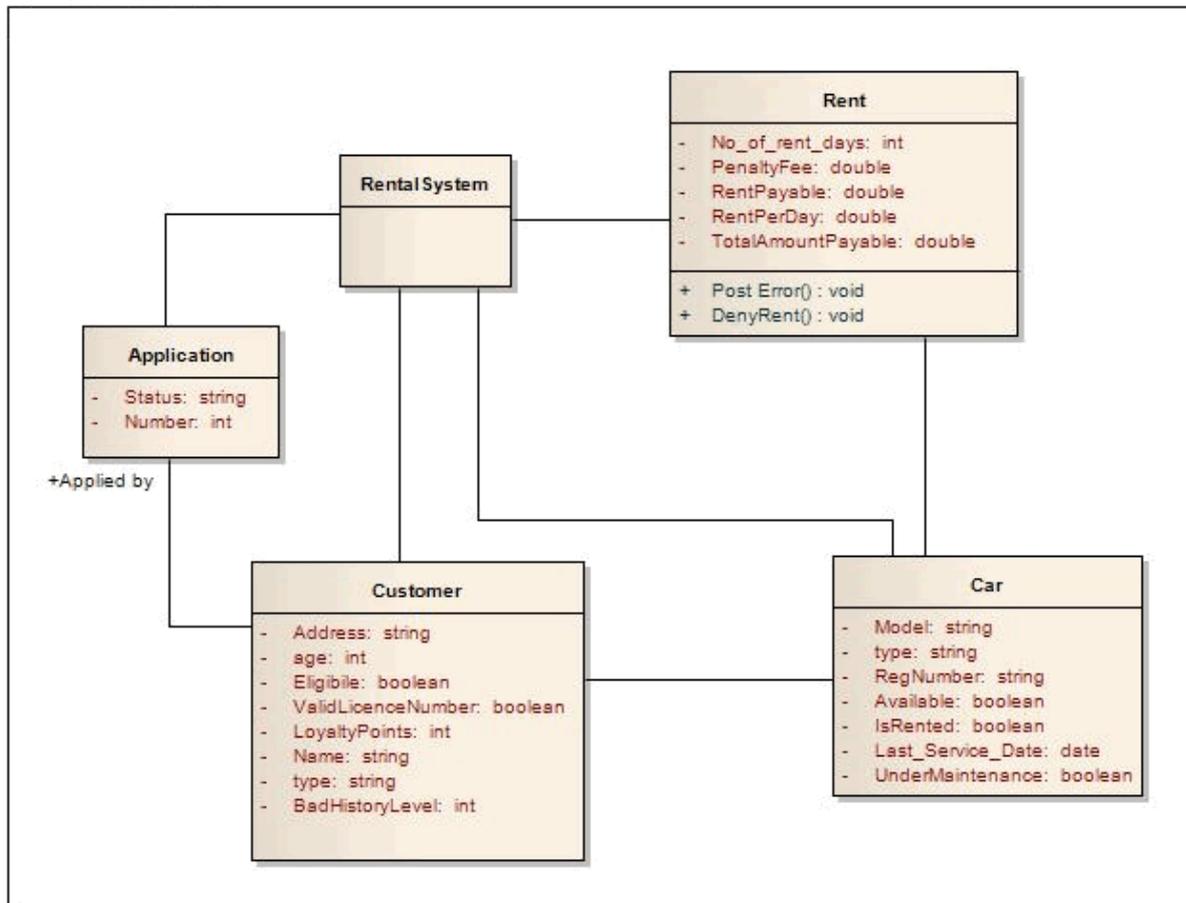
- [Code Generation for Business Rules](#) ^[1217]
- [EASL code templates](#) ^[1137]

8.3.3.2 Create a Business Domain Model

The *Business Domain Model* provides the business vocabulary - terms and facts - on which Business Rules can be modeled. In Enterprise Architect a Business Domain model is created as a conceptual Class diagram.

Example:

The following diagram shows an example Business Domain model, for a Car Rental system.



In the example Business Domain model, the Classes *Rent*, *Customer*, *Car* and *Application*, together with their attributes and operations, provide the terms for the business vocabulary. The Class *Rental System* processes the rules. To make *Rental System* process the rules, you add a **RULE FLOW ACTIVITY** as a behavior for this Class.

When you create a Rule Flow behavior (Activity) under a Class you can model the rules as *Rule Tasks* (Actions). When code is generated the rule flow behavior is rendered as a method inside the corresponding Class.

Alternatively, if you have existing operations in the Class that already suit the purpose, you can **model business rules in those operations**. When code is generated for the Class the rules logic is generated as the method body for the corresponding operation.

Notes:

- When you create Classes in the Business Domain model, select the correct language for code generation to ensure that the correct data type is set for attributes and operation parameters
- Business Rules code generation is supported for the following languages:
 - C++
 - C#
 - Java
 - VB.Net

Learn More:

- [Create a Rule Flow Activity](#)^[1206]
- [Model Rules in an Operation](#)^[1209]

8.3.3.3 Create a Rule Flow Model

You create a *Rule Flow Activity* as a behavior for a Class, to enable that Class to process a set of rules.

To add a Rule Flow Activity to a Class, follow the steps below

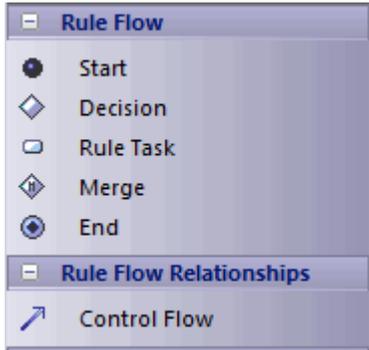
How To:

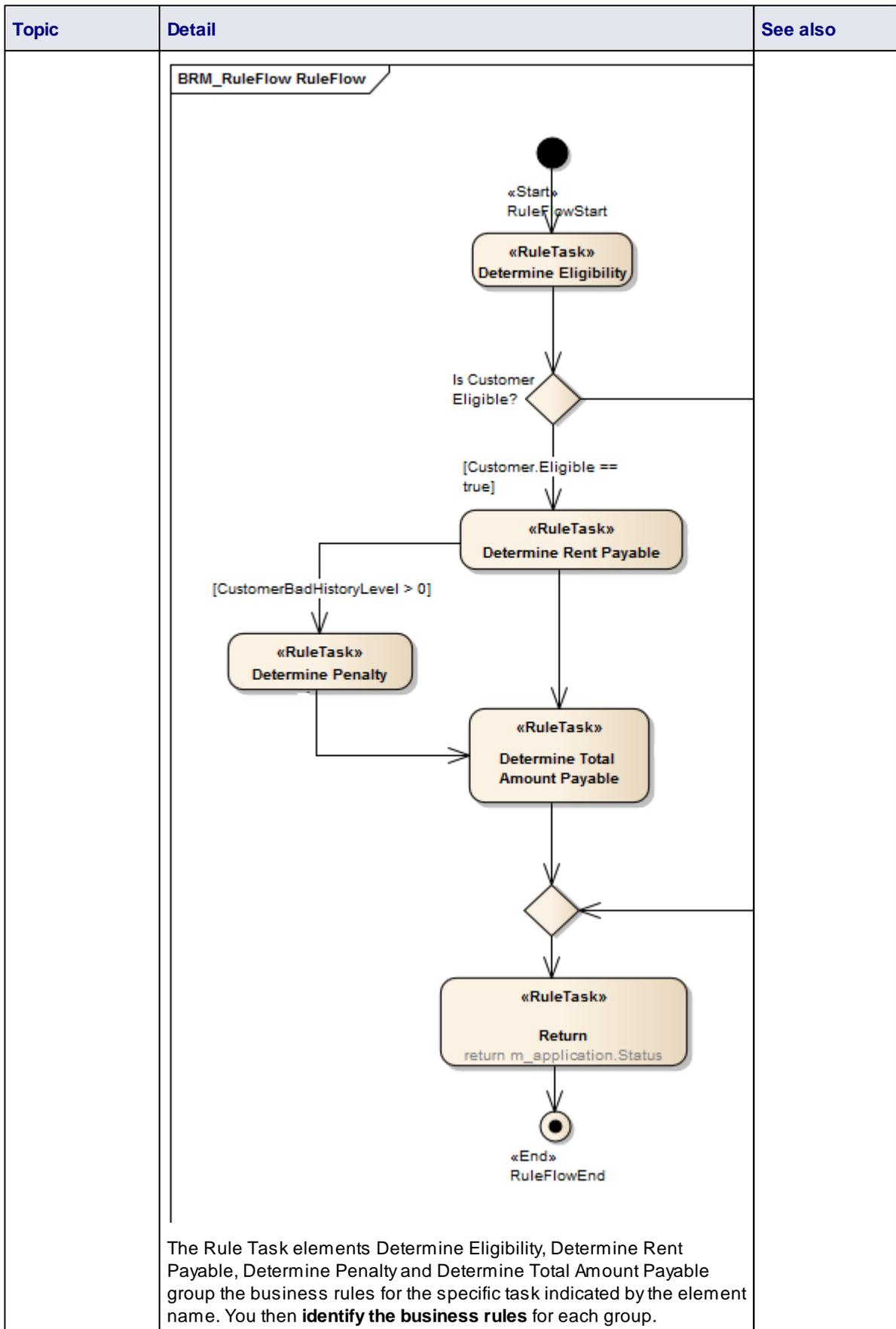
Step	Action	See Also
1	On the Business Domain model diagram, right-click on the Class that processes the rules (in the example, <i>Rental System</i>).	
2	From the context menu select the Add RuleFlow Activity menu option.	

A new Rule Flow Activity with a *Rule Flow diagram* is created as a behavior for the selected Class. The Rule Flow diagram models the sequence in which a series of *Rule Tasks* are executed.

Code generation for a Rule Flow model renders each RuleFlow Activity as a set of operations or methods. Depending on what you want these methods to do, you might want to pass in some parameters to be used within the Rule Flow Activity. See the **Pass Parameters to Rule Flow Activity** topic.

Topics:

Topic	Detail	See also
Add a Rule Task	<p>A Rule Task is a stereotyped Action that groups Business Rules for a specific task. You create Rule Task elements in a Rule Flow diagram using the associated Rule Flow pages of the Toolbox.</p>  <p>The following illustration is of a possible Rule Flow diagram for the car rental example.</p>	Model Business Rules ^[1202]



Notes:

- In a Rule Flow diagram, every Decision Node has a matching *Merge Node* to ensure proper code generation
- For code generation, the Rule Task elements must be grouped inside the appropriate Rule Flow Activity in the **Project Browser**. However, Rule elements can be defined anywhere in the model, as they can be used in more than one Rule Task

Learn More:

- [Pass Parameters to Rule Flow Activity](#)^[1208]

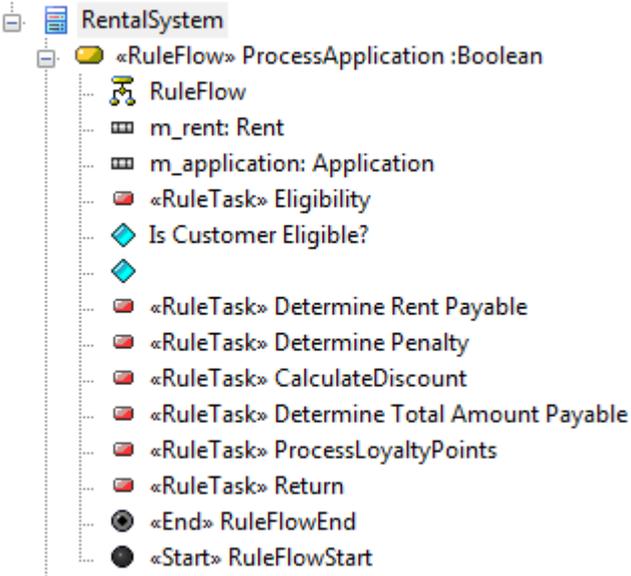
8.3.3.3.1 Pass Parameters to Rule Flow Activity**How To:**

To pass in parameters to be used within a Rule Flow Activity, follow the steps below:

Step	Action	See Also
1	In the Project Browser , double-click on the Rule Flow Activity The element Properties dialog displays	
2	Click on the Behavior tab	
3	Click on the Edit Parameters button The Parameters dialog displays	
4	Create and define each parameter, setting Type and Default values	Parameters Dialog ^[717]
5	Save each parameter and, when you have finished setting the parameters, close both dialogs	

Topics:

Topic	Detail	See also
Rule Flow Activity Parameters	The Rule Flow Activity parameters can be accessed by the Rule Tasks within the parent Rule Flow Activity. In the following hierarchy, the parameters <i>m_rent</i> and <i>m_application</i> can be used by any of the Rule Tasks under the <i>ProcessApplication</i> Rule Flow Activity.	Compose Business Rules ^[1217]

Topic	Detail	See also
	 <p>You can use the parameters as condition variables or action variables in the Business Rule Decision Table, or as rule variables in the Computation Table for any of the Rule Tasks. If the Activity parameter is not accessible to a Rule Task, Enterprise Architect displays an error message.</p>	

8.3.3.3.2 Model Rules In an Operation

You can model business rules either in the **Business Rule** elements attached to the Rule Task element in a **RuleFlow Activity** diagram, or in the operations of the rule Class in the **Business Domain model**.

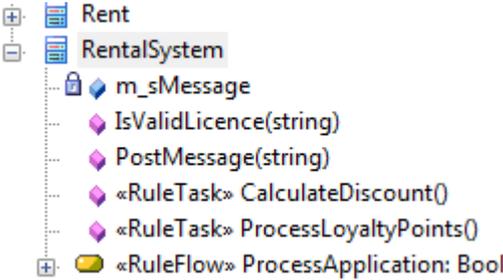
To model business rules for an operation:

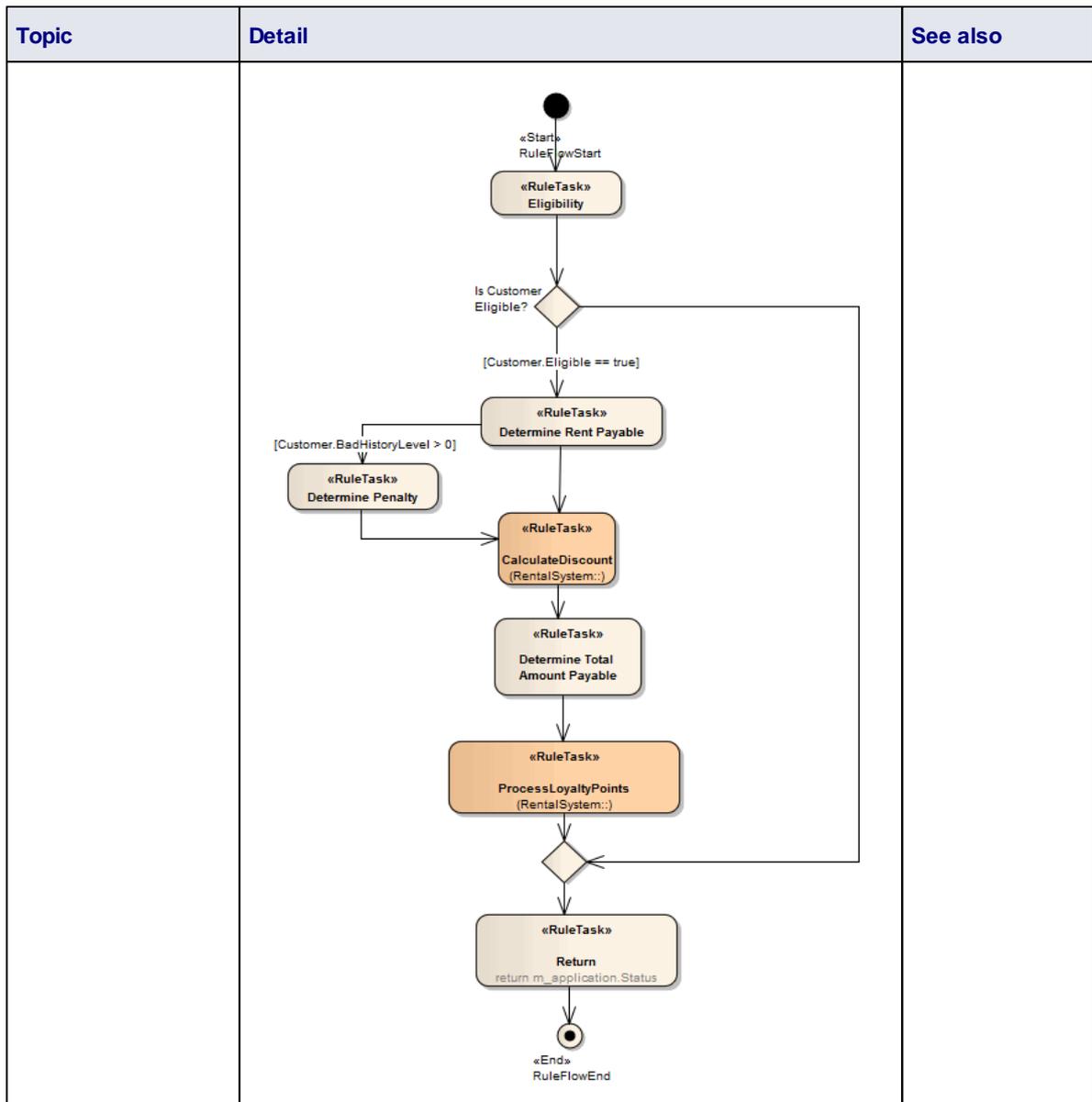
How To:

Step	Action	See Also
1	Open the Properties dialog for the operation and, in the Stereotype field on the General tab, type the value RuleTask	
2	In the Project Browser , right-click on the operation and select the Rule Composer option to open the Rule Composer	
3	Model the rules for the operation	Compose Business Rules 121

Topics:

Topic	Detail	See also
Usage	The operations appear in the Project Browser as shown below:	Behaviour Calls 715

Topic	Detail	See also
	 <p>On code generation, the code for rules logic is generated in the method body.</p> <p>When you drag and drop a RuleTask operation onto a Rule Flow diagram, an operation call behavior action is created. To pass the parameters for this operation call, open the Properties dialog and select the Call tab. Set the Behavior field to the operation to be called. Under the Arguments field, click on the Edit Arguments button and edit the argument values to be passed.</p> <p>On the diagram, the call behavior actions for the RuleTask operations are indicated as shown below:</p>	Behaviour Call Arguments ⁷¹⁶



Learn More:

- [Business Rules](#) ^[1202]
- [RuleFlow Activity Diagram](#) ^[1206]
- [Business Domain Model](#) ^[1204]

8.3.3.4 Compose Business Rules

You use the **Rule Composer** to define a business rule written in plain text within a Business Rule element or Class operation. The **Rule Composer** enables you to model conceptual-level business rules at a logical level in tabulated format, which assists in transforming the rules to technology-specific rules (code).

You can also **download** the contents of the **Rule Composer** to a spreadsheet application such as Microsoft Excel, via a **CSV** file.

Topics:

Topic	Detail	See also
Access The Rule Composer	<p>To access the Rule Composer, right click on a Rule Task element and select the Rule Composer context menu option. The Rule Composer displays in the central work area on its own tab.</p> <p>The Rule Composer consists of:</p> <ul style="list-style-type: none"> • a Rule Statements list • a Decision Table and • a Computation Rule Table <p>To assist with traceability, as the Rule Composer is completed, selections in one table automatically highlight the corresponding rows and columns of the other tables. For example, If a Rule Statement is selected, the related rule column in the Decision Table and row in the Computation Rule Table are highlighted. Similarly, if a Computational Rule is selected, the corresponding column in the Decision Table and row in the Rule Statements list are highlighted.</p>	
Rule Statements Table	<p>The Rule Statements table lists the rules associated with the selected Rule Task. You add a rule to the table by dragging an existing Business Rule element from the Project Browser onto an empty row in the Rule Statements table. You cannot create new rules within the table.</p>	
Decision Table	<p>The Decision Table enables you to model <i>conditional</i> rules (for example: <i>Cars must not be rented to customers of age less than 18</i>).</p> <p>The table has three sections:</p> <ul style="list-style-type: none"> • Rule Conditions – to model condition variables • Rule Actions – to model action variables • Rule Bindings – to link the rule in the rule table 	
Computation Rule Table	<p>The Computation Rule table enables you to model rules involving computations.</p> <p>The table has the following columns:</p> <ul style="list-style-type: none"> • Computation Rule Actions • Expression • Rule Bindings • Rule Dependency 	

How To:

To define a business rule associated with the selected Rule Task, follow the steps below

Step	Action	See Also
1	For the first rule, select the text within the Business Rule element and drag it onto the empty row.	
2	For a subsequent rule, click on the No column and select the Add Row context menu option. An empty row is added to the Rule Statements table.	

Step	Action	See Also
3	Drag the required Business Rule element from the Project Browser onto the new row. If the Business Rule element is not already on the diagram, this adds the element to the diagram and creates a Dependency relationship between the Business Rule and Rule Task elements.	

To remove a rule that is no longer required in the **Rule Composer**, right-click on the appropriate **No** field and select the **Remove Rule** context menu option. This removes the rule from the **Rule Composer** and deletes the Dependency relationship with the Rule Task element. However, it does not remove the Business Rule element from either the diagram or the **Project Browser** (where, in either case, it might be in use with other Rule Task elements).

To model Rule Conditions in the Rule Conditions Section, follow the steps below

Step	Action	See Also
1	The Business Domain model defines the business terms (such as <i>Customer</i>) and their associated attributes. From the appropriate Class element in the Project Browser , drag and drop the required condition attribute (such as age) or operation (such as <i>IsValidLicense()</i>) onto the Rule Conditions column. <ul style="list-style-type: none"> The Rule Condition field enables you to use intellisense to display a list of possible entries for the field. Press (Ctrl+Spacebar) in the field to display the list of entries If the Rule Condition is of type <i>enum</i>, the Allowable Values fields are automatically set with the <i>enum</i> literals. The procedure then ends here 	
2	Define a range of accepted values for the Rule Condition.	
3	Right-click on the Allowable Values column and select the Edit Allowable Values context menu option. The Edit Allowable Values dialog displays. Type each required value or range of values in the Value field, and click on the Save button to display the value in the Allowable Values list box; for example: age could have the values: <18 >18 and <50 >50 <i>IsValidLicense()</i> could return: True False	
4	Click on the OK button to save the values and close the dialog. A new constraint <i>AllowableValues</i> is created for the attribute. <ul style="list-style-type: none"> You can check this constraint by opening the Properties dialog for the attribute and selecting the Constraints tab If the Rule Condition references an enumeration, the enum literals are not editable in the Edit Allowable Values dialog 	
5	If the Rule Condition is an operation, you can pass parameters to it. Right click on the Allowable Values field, and select the Edit Parameters context menu option. The Edit Parameters dialog displays. Select the parameters	

Step	Action	See Also
	<p>and type their values into the Value text box. Click on the OK button to cancel the dialog.</p> <ul style="list-style-type: none"> You can add an operation as a Rule Condition more than once, to allow calling the operation with different sets of parameters. To add another Rule Condition, right-click on the No column and select the Add Row context menu option. An empty row is added to the table. <p>To remove a Rule Condition from the table, right-click on the appropriate No field and select the Delete Row context menu option. This does not affect the original attribute or <i>the new constraint</i> in the model. You can either re-use the attribute with its constraint, or use the attribute Properties dialog to remove the constraint.</p>	

To model Rule Actions in the Rule Actions section, follow the steps below:

Step	Action	See Also
1	<p>In the Rule Actions section, when a specific value of a Rule Condition calls an operation (such as <i>post error</i>) or decision attribute (such as <i>Eligible - Yes/No</i>), you assign the operation or attribute as an action.</p> <p>From a business term Class element in the Project Browser, drag and drop the required attribute or operation onto the Rule Actions field.</p> <p>The Rule Actions field enables you to use intellisense to display a list of possible entries for the field. Press (Ctrl+Spacebar) in the field to display the list of entries.</p>	
2	<p>For an attribute, double-click on the Allowable Values/Parameters field. The Edit Allowable Values dialog displays; type the range of values in the text box (such as Yes, No; or Accept, Reject), click on the Save button and close the dialog. Select the appropriate response in the Result column fields.</p> <p>If the dropped action variable is of type <i>enum</i>, the Allowable Values/Parameters fields are automatically set with the enum literals.</p>	
3	<p>For an operation, a checkbox displays in each of the Result column fields. To call the operation, select the checkbox in the appropriate column.</p> <p>To pass parameters to the operation, double-click on the Allowable Values/Parameters field. The Edit Parameters dialog displays. Select the parameters and type the values into the Value text box. Click on the Save button and close the dialog.</p> <p>You can add an operation as a Rule Action more than once, to allow calling the operation with different sets of parameters.</p>	
4	<p>Click on the Save button in the Rule Composer toolbar to save the values.</p> <p>Alternatively, you can <i>right-click</i> on an Allowable Values/Parameters field to display a context menu with two options:</p> <ul style="list-style-type: none"> If the Rule Action is an attribute, the Edit Allowable Values option is enabled and this displays the Edit Allowable Values dialog If the Rule Action is an operation, the Edit Parameters option is enabled and this displays the Edit Parameters dialog <p>To add another Rule Action, right-click on the No column and select the Add</p>	

Step	Action	See Also
	<p>Row context menu option. An empty row is added to the table.</p> <p>To remove a Rule Action from the table, right-click on the appropriate No field and select the Delete Row context menu option. This does not affect the original attribute or operation in the model.</p>	

To bind a rule in the Rule Bindings Section, follow the steps below

Step	Action	See Also
1	<p>The Rule Bindings section lies on top of the Rule Conditions section. It binds the Rule Condition and Rule Action values to the appropriate rule in the Rule Table.</p> <p>Select the rule number in the Rule Bindings field over one of the Value<n> or Result<n> columns.</p>	
2	<p>Ensure that the values set in the Value<n> or Result<n> field for the Rule Condition or Rule Action, underneath the rule number, all satisfy the rule.</p>	
3	<p>Click on the Save icon in the Rule Composer toolbar.</p> <p>For example, (referring to the screen diagram at the top of this <i>Rule Composer</i> topic) if rule 2 is <i>Car must not be rented to Customers of age less than 18</i>:</p> <ul style="list-style-type: none"> • Select 2 in the Rule field over the Value1 column • Select < 18 against <i>Customer.age</i> in the Value1 column in the Rule Conditions table • Select No against <i>Customer.Eligible</i> in the Result1 column in the Rule Action table • Select Reject against <i>Application.Status</i> in the Result1 column in the Rule Action table 	

To define a computation rule, follow the steps below:

Step	Action	See Also
1	<p>From the Project Browser, drag and drop the appropriate attribute from a Class in the <i>Fact</i> model into the Computation Rule Actions field.</p> <p>Both the Computation Rule Actions field and the Expression field enable you to use intellisense to display a list of possible entries for the fields. Press (Ctrl+Spacebar) in the field to display the list of entries.</p>	
2	<p>In the Expression field, type the expression to be evaluated.</p>	
3	<p>In the Rule Bindings field, type the rule number from the Rule table of the rule being modeled, to link the table data to the rule.</p>	
4	<p>If the rule depends on another rule being satisfied first, type the number of that rule in the Rule Dependency field.</p>	
5	<p>Click on the Save icon in the Rule Composer toolbar to save the computation rule.</p>	

Step	Action	See Also
	If the computation rule is also a Rule Conditions rule, add the condition variable in the Decision table and bind the appropriate rule in the Rule Bind section.	

To export the contents of the Rule Composer to a CSV file, follow the steps below

Step	Action	See Also
1	Click on the Export to CSV icon () in the Rule Composer toolbar. The Windows Browser dialog displays.	
2	Browse to the required file location and type in a .CSV filename to export to.	
3	Click on the Save button to export the data.	

Learn More:

- [Code Generation for Business Rules](#) ^[1217]
- [CSV Import and Export](#) ^[337]

8.3.3.5 Validate Business Rules

Topics:

Topic	Detail	See also
Usage	<p>It is recommended practice to validate the business rules in the Rule Composer before you generate code for the Rule Task elements. To do this, click on the Validation (green tick) icon in the Rule Composer toolbar.</p>  <p>The business rules on the Rule Composer are parsed and any errors or warnings that might indicate incomplete or unfavorable code generation are displayed on a Rule Composer Validation tab on the Output screen.</p> <p>To highlight and investigate the faulty data in the Rule Composer, double-click on the appropriate warning or error message.</p>	

8.3.3.6 Code Generation For Business Rules

After you have modeled the business rules for all the Rule Task elements in the Rule Flow diagram, you can generate code from the Rule Flow behavior.

How To:

To return a value from the Rule Flow behavior, follow the steps below:

Step	Action	See Also
1	Double-click on the last Rule Task element before the end node of the Rule Flow diagram. The element's Properties dialog displays.	
2	Click on the Effect tab.	
3	In the Effect field, type the <i>return</i> statement; for example, <i>return true</i> .	
4	Click on the Save button, and on the OK button to close the dialog. Generate code for the Class containing the rule flow behavior (in our initial example, Rental System). The code for business rules logic is generated, with the rule statements expressed in natural language as comments.	Generate code for a single class ^[149] Create a Business Domain Model ^[120]

Example:

The following code snippet was generated from the *Rental System* Class element in our example:

```

////////////////////////////////////
// RentalSystem.cs
// Implementation of the Class RentalSystem
// Generated by Enterprise Architect
// Created on:      08-May-2009 2:39:23 PM
////////////////////////////////////

public class RentalSystem {

    public Customer m_Customer;
    public Car m_Car;
    public Rent m_Rent;

    public RentalSystem() {

    }

    ~RentalSystem() {

    }

    public virtual void Dispose() {

    }

    /* Begin - EA generated code for Activities and Interactions */
    public bool ProcessApplication(Rent m_rent, Application m_application)
    {
        // behavior is an Activity

        /* CAR MUST NOT BE RENTED TO CUSTOMERS W THOUT A VALID

```

```

LICENCE NUMBER*/
    if( m_Customer.ValidLicenceNumber == "FALSE" )
    {
        m_application.Status = "Reject";
        m_Customer.Eligible = false;
    }
/* CAR MUST NOT BE RENTED TO CUSTOMERS OF AGE LESS THAN
18*/
    if( m_Customer.age < 18 )
    {
        m_application.Status = "Reject";
        m_Customer.Eligible = false;
    }
/* CAR MUST NOT BE RENTED TO CUSTOMERS WITH BAD HISTORY
LEVEL 3*/
    if( m_Customer.BadHistoryLevel == 3 )
    {
        m_application.Status = "Reject";
        m_Customer.Eligible = false;
    }
    if ( Customer.Eligible == true)
    {
        /* RENT FOR SMALL CARS IS 80 AUD PER DAY*/
        if( m_Car.type == Small )
        {
            m_rent.Rent Per Day = 80;
        }
        /* RENT FOR AWD CARS IS 100 AUD PER DAY*/
        if( m_Car.type == AWD )
        {
            m_rent.Rent Per Day = 100;
        }
        /* RENT FOR LUXURY CARS IS 150 AUD PER DAY*/
        if( m_Car.type == Luxury )
        {
            m_rent.Rent Per Day = 150;
        }
        /* RENT PAYABLE IS CALCULATED AS THE PRODUCT OF
RENTPERDAY AND RENTALPERIOD IN DAYS*/
        m_rent.Rent Payable = m_rent.Rent Per Day * m_rent.
No_of_rent_days;
        if ( Customer.BadHistoryLevel > 0)
        {
            /* PENALTY OF 20 % OF RENT MUST BE APPLIED FOR
CUSTOMERS WITH BAD HISTORY LEVEL 2*/
            if( m_Customer.BadHistoryLevel == 2 )
            {
                m_rent.PenaltyFee = m_rent.Rent Payable
* 0.2;
            }
            /* PENALTY OF 10 % OF RENT MUST BE APPLIED FOR
CUSTOMERS WITH BAD HISTORY LEVEL 1*/
            if( m_Customer.BadHistoryLevel == 1 )
            {
                m_rent.PenaltyFee = m_rent.Rent Payable
* 0.1;
            }
        }
    }
else
{
}

/* TOTAL AMOUNT PAYABLE IS CALCULATED AS THE SUM OF
RENT PAYABLE AND PENALTY IF ANY.*/
    m_rent.Total Amount Payable = m_rent.Rent Per Day +
m_rent.PenaltyFee;
}
else

```

```

    {
    }
    return m_application.Status;
}

/* End - EA generated code for Activities and Interactions */

} // end Rental System

```

8.3.4 BPMN Models

The BPMN notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles integrated with the Enterprise Architect installer. Through use of these profiles, Enterprise Architect enables you to develop BPMN diagrams quickly and simply.

The Enterprise Architect installer for releases later than 8.0 provides you with separate versions of the MDG Technology for BPMN that support BPMN versions 1.0, 1.1 and 2.0, and BPEL.

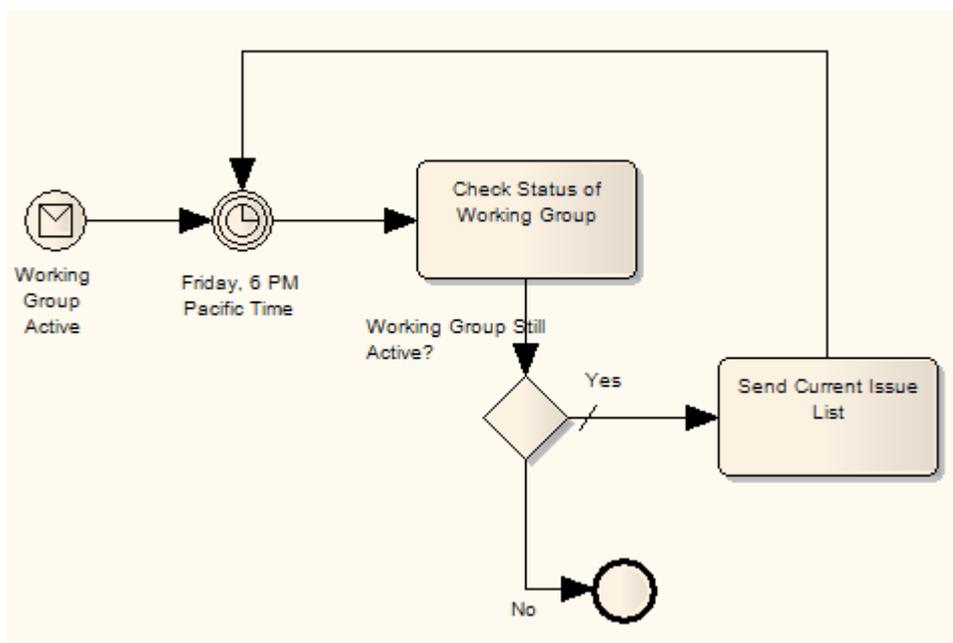
Use To:

- Maintain existing diagrams created in BPMN 1.0 format, and create new diagrams in BPMN 1.0
- Create and maintain diagrams in the BPMN 1.1 and BPEL formats
- Create and maintain diagrams in the BPMN 2.0 and BPEL formats
- Migrate a BPMN 1.0 model (or part of a model) to BPMN 1.1

Access: The BPMN facilities are provided in the form of:

- A BPMN diagram type, accessed through the New Diagram dialog
- BPMN pages in the Toolbox
- BPMN element and relationship entries in the Toolbox **Shortcut** menu and Quick Linker

Specifications of BPMN elements and relationships are defined by Tagged Values (for example, to define the *Message*, *Timer* and *Default Path (/)* symbols in the diagram below).



BPMN Toolbox Pages:

You can access the BPMN Toolbox pages through the **View | Toolbox: More tools...** option; select the **BPMN 1.0, 1.1** or **2.0** options as appropriate.

You can also set BPMN as the active default technology to access the Toolbox pages directly.

Disable BPMN:

If you prefer not to use BPMN in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

Learn More:

- [BPMN 1.0 and 1.1 Toolbox Pages](#) ^[1220]
- [BPMN 2.0 Toolbox Pages](#) ^[1222]
- [BPMN 2.0 XML](#) ^[1234]
- [Migrate a BPMN 1.0 Model to BPMN 1.1](#) ^[1232]
- [Migrate a BPMN 1.1 Model to BPMN 2.0](#) ^[1235]
- [Change BPMN Element Appearance](#) ^[1228]
- [BPEL Models](#) ^[1235]
- [New Diagram dialog](#) ^[568]
- [Set Technology as Active Default](#) ^[1034]

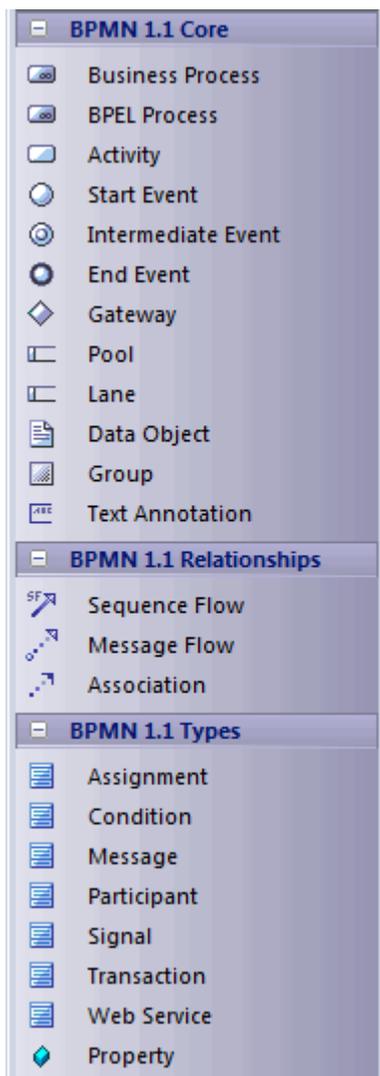
8.3.4.1 BPMN 1.0 and 1.1 Toolbox Pages

The BPMN 1.0 and BPMN 1.1 pages of the Diagram Toolbox enable you to create graphical (Core) and non-graphical (Types) BPMN elements and relationships on business process diagrams in BPMN 1.0, BPMN 1.1 and BPEL formats.

Access: **View | Toolbox: More tools | BPMN 1.0**
View | Toolbox: More tools | BPMN 1.1

The elements and relationships available are shown below, and described in the following table.

Example:



Page	Item	Action
Core	Business Process	Extend a <i>composite Activity</i> that defines a business process
	BPEL Process	Define the behavior of an executable or abstract business process
	Activity	Define an activity within a business process
	Start Event	Define the initiating event in a process
	Intermediate Event	Define an intermediate event in a process
	End Event	Define the terminating event in a process
	Gateway	Define a decision point in a business process If a condition is true , then processing continues one way; if not, then another
	Pool	Extend a <i>Partition</i> element to logically organize an Activity
	Lane	Extend a <i>Partition</i> element to subdivide a Pool

Page	Item	Action
	Data Object	Extend an <i>Artifact</i> element to define a physical piece of information used or produced by a system
	Group	Extend a <i>Boundary</i> element to group other elements
	Text Annotation	Create a comment
Relationships	Sequence Flow	Extend a <i>Control Flow</i> relationship to define the flow of activity
	Message Flow	Extend a <i>Control Flow</i> relationship to define the flow of communications in the process
	Association	Associate information and artifacts with flow objects
Types	Assignment	Define the <i>properties</i> (Tagged Values) of the Core BPMN 1.1 elements such as Activities, Events and Gates
	Condition	
	Message	
	Participant	
	Signal	
	Rule	
	Transaction	
	Web Service	
	Property	

8.3.4.2 BPMN 2.0 Toolbox Pages

The BPMN 2.0 pages of the Diagram Toolbox enable you to create BPMN elements and relationships on business process diagrams in BPMN 2.0 and BPEL formats.

Access: [View](#) | [Toolbox: More tools](#) | [BPMN 2.0](#)

Enterprise Architect provides a set of pages for each of the following BPMN 2.0 diagram types:

Topic	Link
Business Process - contains the constructs needed to build Business Process models (and, effectively, Collaboration models)	Business Process ^[1222]
Choreography - contains the constructs needed to build Choreography models	Choreography ^[1224]
Conversation - contains the constructs needed to build Conversation models	Conversation ^[1225]
Types - contains the constructs common to all BPMN 2.0 diagrams	Type ^[1227]

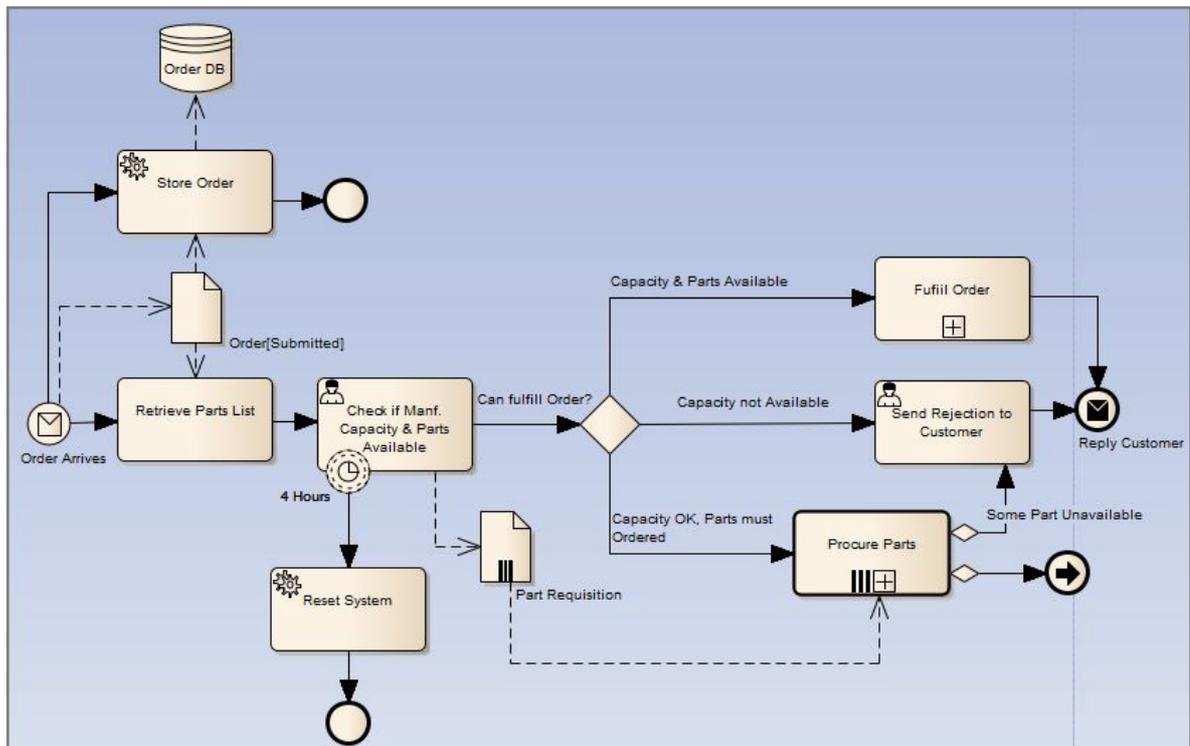
8.3.4.2.1 BPMN 2.0 Business Process Toolbox Pages

The BPMN 2.0 Business Process pages of the Diagram Toolbox enable you to create BPMN elements and relationships on business process diagrams in BPMN 2.0 and BPEL formats.

Access: [View](#) | [Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Business Process](#)

Page	Item	Use to
BPMN 2.0 Business Process	Business Process	Extend a composite Activity that defines a business process
	BPEL	Define the behavior of an executable or abstract business process
	Activity	Define an activity within a business process
	Choreography	Extend an Activity element to represent a process unit of information exchange between Participants Enterprise Architect currently supports two to four participant bands within one Choreography Activity/Task
	Conversation	Extend a Class element to group a set of Message Flows together based on a certain concept
	Data Object	Provide or store the information for an Activity
	Data Store	Represent a mechanism for an Activity to retrieve or update stored information
	Start Event	Define the initiating event in a process
	Intermediate Event	Define an intermediate event in a process
	End Event	Define the terminating event in a process
	Gateway	Define a decision point in a business process If a condition is true then processing continues one way; if false , then another
	Pool	Extend a Partition element to logically organize an Activity
	Lane	Extend a Partition element to subdivide a Pool
	Message	Represent the contents of a communication between two Participants
	Group	Extend a Boundary element to group other elements
Text Annotation	Create a comment	
BPMN 2.0 Business Process Connectors	Sequence Flow	Extend a Control Flow relationship to define the flow of activity
	Message Flow	Extend a Control Flow relationship to define the flow of communications in the process
	Data Association	Move data between Data Objects, Data Store, Properties and Activities, Processes
	Association	Link the information and Artifacts with BPMN graphic elements
	Conversation Link	Connect a Conversation Node with a Participant, in either direction

Example of a BPMN 2.0 Business Process Diagram:



Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

Learn More:

- [Change BPMN Element Appearance](#) ^[1226]

8.3.4.2.2 BPMN 2.0 Choreography Toolbox Pages

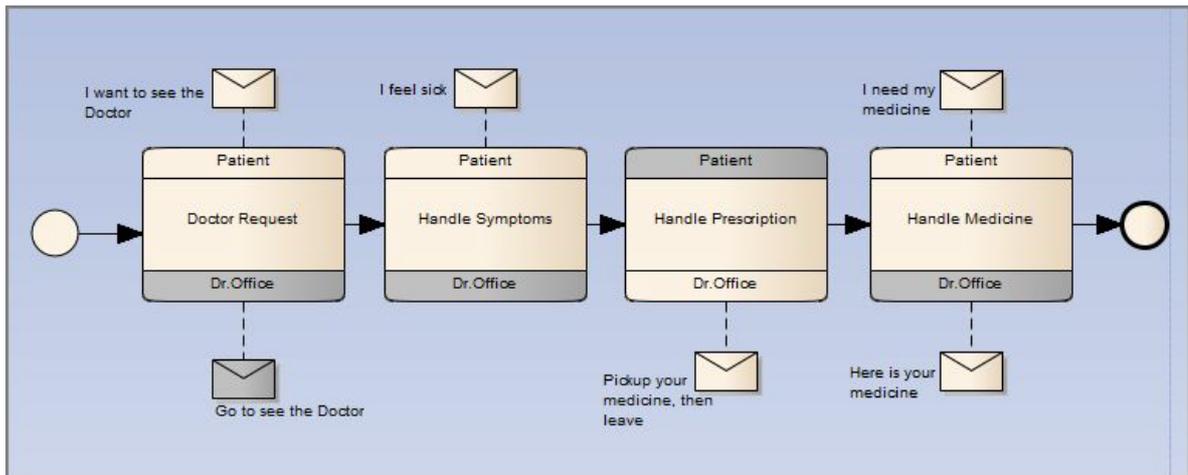
The BPMN 2.0 Choreography pages of the Diagram Toolbox enable you to create BPMN elements and relationships on Choreography diagrams in BPMN 2.0 and BPEL formats.

Access: [View](#) | [Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Choreography](#)

Page	Item	Use to
BPMN 2.0 Choreography	Choreography Model	Extend a composite Activity that defines a Choreography process
	Choreography	Extend an Activity element to represent a process unit of information exchange between Participants Enterprise Architect currently supports two to four participant bands within one Choreography Activity/Task
	Start Event	Define the initiating event in a process
	Intermediate Event	Define an intermediate event in a process
	End Event	Define the terminating event in a process

Page	Item	Use to
	Gateway	Define a decision point in a business process If a condition is true then processing continues one way; if false , then another
	Message	Represent the contents of a communication between two Participants
	Pool	Extend a Partition element to logically organize an Activity
	Text Annotation	Create a comment
BPMN 2.0 Choreography Connectors	Sequence Flow	Define the order of activity in a Choreography
	Association	Link the information and Artifacts with BPMN graphic elements
	Message Flow	Extend a Control Flow relationship to define the flow of communications in the process

Example of a BPMN 2.0 Choreography Diagram:



Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

Learn More:

- [Change BPMN Element Appearance](#)^[1228]

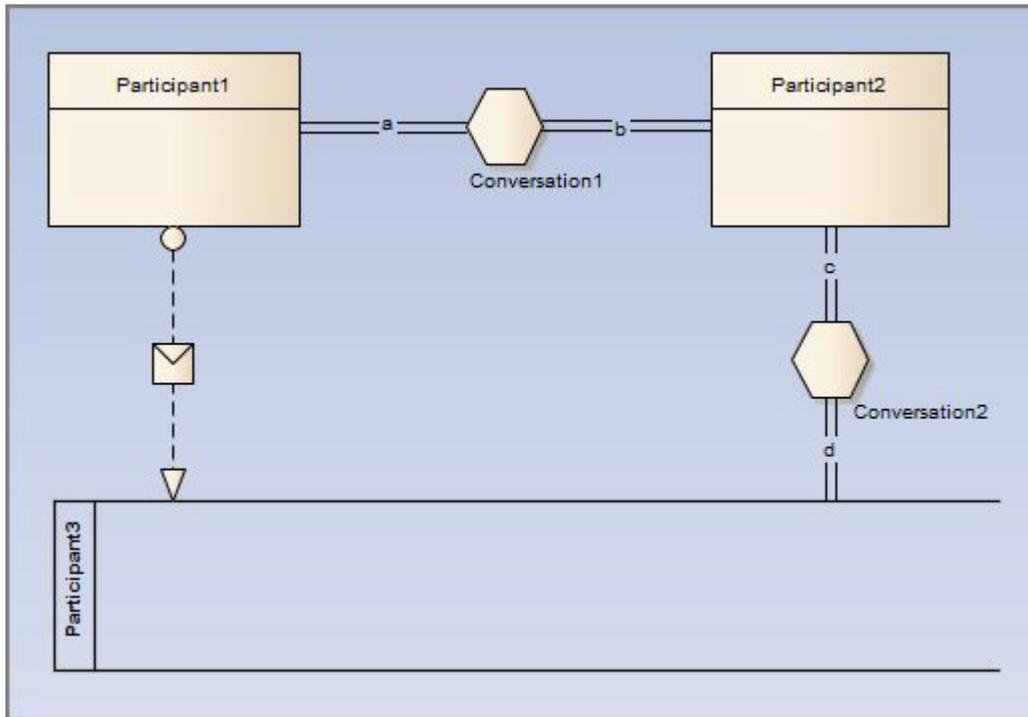
8.3.4.2.3 BPMN 2.0 Conversation Toolbox Pages

The BPMN 2.0 Conversation pages of the Diagram Toolbox enable you to create BPMN elements and relationships on Conversation diagrams in BPMN 2.0 and BPEL formats.

Access: [View](#) | [Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Conversation](#)

Page	Item	Use to
BPMN 2.0 Conversation	Conversation Model	Extend a composite Activity that defines the logical relationship of Message exchanges between two or more business entities
	Conversation	Extend a Class element to group a set of Message Flows together based on a certain concept
	Participant	Represent a specific organization or individual that is involved in a business process
	Pool	Extend a Partition element to logically organize an Activity
	Activity	Define an activity within a business process
	Start Event	Define the initiating event in a process
	Intermediate Event	Define an intermediate event in a process
	End Event	Define the terminating event in a process
	Gateway	Define a decision point in a business process If a condition is true then processing continues one way; if false , then another
	Text Annotation	Create a comment
BPMN 2.0 Conversation Connectors	Conversation Link	Connect a Conversation Node to or from a Participant
	Message Flow	Extend a Control Flow relationship to define the flow of communications in the process
	Sequence Flow	Extend a Control Flow relationship to define the flow of activity
	Association	Link the information and artifacts with BPMN graphic elements

Example of a BPMN 2.0 Conversation Diagram:



Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

Learn More:

- [Change BPMN Element Appearance](#) 

8.3.4.2.4 BPMN 2.0 Type Toolbox Page

The BPMN 2.0 Types page of the Diagram Toolbox is common to all BPMN 2.0 diagrams.

Access: [View](#) | [Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 <diagram type>](#)

Page	Item	Use To
BPMN 2.0 Types	Assignment	Define the non-graphic elements or properties (Tagged Values) of the Core BPMN 2.0 graphic elements
	Correlation Key	
	Correlation Property	
	Correlation Property Binding	
	Correlation Property Retrieval	
	Correlation Subscription	
	Complex Behavior Definition	

Page	Item	Use To
	Item Definition	
	Error	
	Escalation	
	Signal	
	IO Specification	
	Input Set	
	Output Set	
	Input Output Binding	
	Interface	
	Operation	
	Participant	
	Resource	
	Resource Role	
	Resource Parameter	
	Property	
	Participant Association	
	Conversation Association	

8.3.4.3 Change BPMN Element Appearance

A number of Tagged Values directly affect the appearance of the BPMN element or relationship they apply to. Therefore, by changing the value of the tag you can change the appearance of the object.

Access: [View | Tagged Values](#)

How To:

To define the appearance of BPMN elements and relationships, follow the steps below:

Step	Action	See Also
1	Open the Tagged Values window	Tagged Values [762]
2	Select the required element or relationship in a diagram The Tagged Values window shows the appropriate Tagged Values and provides a list of values to assign to each one	
3	Set the required value, based on the value descriptions in the table below	

Reference:

Element	Tags	Values Setting
Activity	<i>activityType</i>	Set to: <ul style="list-style-type: none"> • Task for no decoration • Sub-Process to display the <i>plus-in-box</i> decoration on the bottom edge of the shape
	<i>adHoc</i>	Set to true to display the <i>tilde</i> decoration on the bottom edge of the shape, indicating that this is an ad-hoc Activity
	<i>adHocOrdering</i>	Set to: <ul style="list-style-type: none"> • Parallel to display three vertical lines on the bottom edge of the shape • Sequential to display three horizontal lines on the bottom edge of the shape • Unspecified to display no lines on the bottom edge of the shape The <i>adHoc</i> tag must first be set to true
	<i>isACallActivity</i>	Set to true to show the boundary of the Activity element as a thick line, indicating that this is a Call Activity
	<i>isATransaction</i>	Set to true to give the Activity element a double-lined border, indicating that this is a Transaction
	<i>isForCompensation</i>	Set to true to display the <i>rewind</i> icon on the bottom edge of the shape, indicating that this is a Compensation Activity
	<i>isSequential</i>	Set to true to display three horizontal bars on the bottom edge of the shape The <i>loopCharacteristics</i> tag must first be set to Multinstance
	<i>loopCharacteristics</i>	Set to: <ul style="list-style-type: none"> • Standard to display three vertical bars at the bottom edge of the shape • Multinstance to display a loop icon at the bottom edge of the shape
	<i>taskType</i>	When set to different value, the associated decoration is displayed at the top-left corner of the shape
<i>triggeredByEvent</i>	Set to true to show the boundary of the Activity element as a dotted line	
Choreography	<i>choreographyType</i>	Set to: <ul style="list-style-type: none"> • Task for no decoration • Sub-Process to display the <i>plus-in-box</i> decoration on the bottom edge of the shape
	<i>initiatingParticipant</i>	Set a value to show the corresponding participant band in a light color, which indicates that this is the initiating participant
	<i>isACallChoreography</i>	Set to true show the boundary of the Choreography element as

Element	Tags	Values Setting
		a thick line
	<i>loopType</i>	Set to: <ul style="list-style-type: none"> • Standard to display three vertical bars at the bottom edge of the shape • Multilinstance to display a loop icon at the bottom edge of the shape Each Participant has its own Multilinstance setting
	<i>numberOfParticipants</i>	Set to the appropriate numerical value to indicate the number of Participant bands for a choreography Activity or Task
	<i>participantA</i> or <i>participantARef</i>	<i>participantA</i> - Type the name of the Participant to show on the Participant band of the Choreography If the Participant is known and referenced, use the <i>participantARef</i> tag and leave the <i>participantA</i> tag blank; the name of the referenced Participant is shown on the Participant band of the Choreography The same usage is applied to other Participant bands
Conversation	<i>isACallConversation</i>	Set to true to show the boundary of the Conversation element as a thick line
	<i>isComposite</i>	Set to true to display a <i>plus-in-box</i> decoration on the bottom edge of the shape, indicating that the Conversation is a composite element
Data Object	<i>dataInOut</i>	Set to: <ul style="list-style-type: none"> • Input to add a light-colored arrow at the top-left corner of the Data Object, indicating that it represents a Data Input element • Output to add a dark-colored arrow at the top-left corner of the Data Object, indicating that it represents a Data Output element
	<i>isCollection</i>	Set to true to add three vertical lines at the bottom edge of the shape, indicating that the object is a collection
Event	<i>cancelActivity</i>	For Intermediate Event only Set to false to show the Intermediate Event border as a dashed line
	<i>catchOrThrow</i>	For Intermediate Event only Set to: <ul style="list-style-type: none"> • Catch or • Throw
	<i>eventDefinition</i>	Use to change the decoration of a Start Event, Intermediate Event or End Event
	<i>isInterrupting</i>	For Start Event only Set to false to show the border of the Start Event as a dashed line, indicating that the Start Event is not an interrupting event

Element	Tags	Values Setting
Gateway	<i>eventGatewayType</i>	Use to define two types of Event-based Gateway: <ul style="list-style-type: none"> • Exclusive and • Parallel
	<i>gatewayType</i>	Set to: <ul style="list-style-type: none"> • Exclusive to render the Gateway as a diamond shape without any marker in the middle; if you also set the <i>markerVisible</i> tag to true, the Gateway is rendered as a diamond shape with an X inside • Complex to render the Gateway as a diamond shape with an asterisk (*) inside • Inclusive to render the Gateway as a diamond shape with a circle (o) inside • Parallel to render the Gateway as a diamond shape with a plus (+) inside • Event to define the Gateway as Event-based
	<i>instantiate</i>	Use only to define the Exclusive and Parallel Event-based Gateway
Message	<i>IsInitiate</i>	Set to true to make the envelope color gray, indicating that this is an initiating Message
Message Flow	<i>messageVisible</i>	<ul style="list-style-type: none"> • Set to Unspecified to make the Message Flow connector without decoration • Set to Initiating to add a white envelope to the Message Flow connector • Set to Non-Initiating to add a gray envelope to the Message Flow connector
Participant	<i>participantMultiplicity</i>	Set to true to display three vertical bars at the bottom edge of a rectangle Pool element. Participant is rendered as a normal UML Class shape exclusively on Conversation diagrams
Pool	<i>blackBoxPool</i>	Set to true to render the Pool element as a rectangle, which is used in a Main Pool diagram
	<i>participantMultiplicity</i>	Set to true to display three vertical bars at the bottom edge of a rectangle Pool element
Sequence Flow	<i>conditionType</i>	<ul style="list-style-type: none"> • Set to Expression to show an unfilled diamond marker at the source end of the connector • Set to Default to show a slash marker at the source end of the connector

Notes:

- Some BPMN elements changed in appearance between BPMN version 1.0 and BPMN version 1.1
- In Enterprise Architect releases later than 7.1, if you work on a BPMN 1.0 model created in an earlier release, existing elements default to their version 1.0 appearance; new elements assume the BPMN version 1.1 appearance and automatically have a Tagged Value BPMNVersion set to 1.1
- If you want a new element to revert to the BPMN version 1.0 appearance, set the BPMNVersion Tagged Value to 1.0
- Conversely, if you want an older element to assume the BPMN version 1.1 appearance, assign the BPMNVersion Tagged Value to it, with the value 1.1

8.3.4.4 Migrate BPMN 1.0 Model to BPMN 1.1

Enterprise Architect enables you to migrate a BPMN 1.0 model (or part of a model) to BPMN 1.1 or BPMN1.1::BPEL, using the Automation Interface. There are two functions you can use:

- *MigrateToBPMN11()*
 - `proj.MigrateToBPMN11 sGUID, "BPMN"`
 - `proj.MigrateToBPMN11 sGUID, "BPEL"`
- *Migrate()*
 - `proj.Migrate sGUID, "BPMN", "BPMN1.1"`
 - `proj.Migrate sGUID, "BPMN", "BPMN1.1::BPEL"`

These functions update the Tagged Values and, if required, stereotypes to BPMN 1.1 for all elements, attributes, connectors and diagrams under the selected package or element.

Example:

The following VB script calls the *MigrateToBPMN11()* function to migrate the Tagged Values to BPMN 1.1:

```
Sub MigrateElement (sGUID, lngPackageID)

    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.MigrateToBPMN11 sGUID, "BPMN"

    'refresh the model
    If lngPackageID <> 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If

End Sub

Sub MigrateSelectedItem

    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package

    selType = GetTreeSelectedItemType

    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Complete", 0, "BPMN 1.1 Migration"

    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Complete", 0, "BPMN 1.1 Migration"

    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "BPMN 1.1 Migration"

    End If

End Sub

Sub Main

    MigrateSelectedItem

End Sub

Main
```

Notes:

- In BPMN 1.0, various tags have free-text direct-entry value fields, and you can provide additional information on these tags in the Tagged Value Note dialog, for display at the bottom of the Tagged Values window

In BPMN 1.1, some of these tags (such as the *Categories* tag on a *BusinessProcess* stereotyped element) have been changed to memo type, and you use the Tagged Value Note dialog to enter the value; therefore, you cannot have additional notes for these tags, all information must be within the tag's value

- For such tags, when migrating from BPMN 1.0 to BPMN 1.1, the BPMN 1.0 tag value is moved into the BPMN 1.1 tag notes field and the BPMN 1.0 tag notes are discarded; if you want to preserve the tag notes text, take a copy of the BPMN 1.0 model before migration to enable you to copy the tag notes text into the tag value after migration

Learn More:

- [MigrateToBPMN11\(\)](#)^[1942]
- [Migrate\(\)](#)^[1942]

8.3.4.5 Migrate BPMN 1.1 Model to BPMN 2.0

Enterprise Architect enables you to migrate a BPMN 1.1 model (or part of a model) to BPMN 2.0, using the Automation Interface function *Migrate()*. This function updates the Tagged Values and, if required, stereotypes to BPMN 2.0 for all elements, attributes, connectors and diagrams under the selected package or element.

Example:

The following VB script calls the *Migrate()* function to migrate the package or element to BPMN 2.0:

```
Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "BPMN1.1", "BPMN2.0"

    'refresh the model
    If lngPackageID > 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub

Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package
    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed", 0, "BPMN 2.0 Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Package Migration Completed", 0, "BPMN 2.0 Migration"
    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "BPMN 2.0 Migration"
    End If
End Sub

Sub Main
```

```

    MigrateSelectedItem
End Sub

```

```

Main

```

Notes:

- Please backup your project before you run the BPMN 2.0 Migrator
- Normalization occurs on the following tags:
 - BPMN1.1::Activity::InMessageRef
 - BPMN1.1::Activity::OutMessageRef
 - BPMN1.1::Activity::IORules
 - BPMN1.1::Activity::InputSets
 - BPMN1.1::Activity::OutputSets
 - BPMN1.1::Activity::ComplexML_FlowCondition
 - BPMN1.1::Activity::Performers
 - BPMN1.1::BusinessProcess::InputSets
 - BPMN1.1::BusinessProcess::OutputSets
 - BPMN1.1::BusinessProcess::Performers
 - BPMN1.1::EndEvent::ErrorCode
 - BPMN1.1::IntermediateEvent::ErrorCode

Taking *BPMN1.1::Activity::InMessageRef* as an example, the following steps take place:

- (1) Create a new element *BPMN2.0::Operation*
- (2) Insert the reference tag *BPMN2.0::Activity::operationRef*
- (3) Migrate *BPMN1.1::Activity::InMessageRef* to *BPMN2.0::Operation::InMessageRef*

- Denormalization occurs on the following tags:
 - BPMN1.1::Transaction::TransactionProtocol
 - BPMN1.1::WebService::Operation
 - BPMN1.1::WebService::Interface
 - BPMN1.1::WebService::ParticipantRef
 - BPMN1.1::Condition::ConditionExpression
 - BPMN1.1::BPELProcess::InputSets
 - BPMN1.1::BPELProcess::OutputSets

Taking *BPMN1.1::Transaction::TransactionMethod* as an example, the following steps take place:

- (1) Find element *BPMN1.1::Transaction* by '*BPMN1.1::Transaction::TransactionProtocol*
- (2) Migrate *BPMN1.1::Transaction::TransactionMethod* to *BPMN2.0::transactionMethod*
- (3) Migrate *BPMN1.1::Transaction::TransactionProtocol* to *BPMN2.0::transactionProtocol*

- If a BPMN1.1 stereo tag is discarded in BPMN2.0, it is preserved if the tag has a value

Learn More:

- [Migrate\(\)](#)^[1942]

8.3.4.6 BPMN 2.0 XML

Enterprise Architect supports serializing **BPMN 2.0 Models** in **BPMN 2.0 XML**. The serialized XML file contains both the model semantics as well as the diagram-interchange information.

Access: Click on a package in the Project Browser, then:

Project | Model Publisher

**Project | Model Import/Export | Export Package to XMI File: Publish, or
Right-click | Import/Export | Export Package to XMI: Publish**

How To:

Step	Action	See Also
1	Open the Publish Model Package dialog	Publish Model Package [32]
2	In the XML Type field, select BPMN 2.0 XML	
3	Click on the Export button to initiate the BPMN 2.0 XML serialization	

Notes:

1. The package being serialized is assumed to be self-contained
2. Only the contents of the selected package are considered for serialization - child packages (and their contents) are ignored
3. Serialization of the *Group* element in the BPMN 2.0 Business Process Tool-box is not supported
4. Serialization of the following items in the BPMN 2.0 Types section of the Tool-box is not supported :
 - *InputOutputBinding*
 - *Participant Association*
 - *Conversation Association*

Learn More:

- [XML Schema for BPMN 2.0 XML](#) (Online Resource)
- [BPMN 2.0 Toolbox Pages](#) [1222]

8.3.5 BPEL Models

The following text is derived from the **BPEL** entry in the online Wikipedia:

Business Process Execution Language (BPEL), short for **Web Services Business Process Execution Language (WS-BPEL)**, is an executable language for specifying interactions with **Web Services**. Processes in Business Process Execution Language export and import information by using Web Service interfaces exclusively.

Web service interactions can be described in two ways :

1. Executable business processes, which model the actual behavior of a participant in a business interaction.
2. Abstract business processes, which are partially specified processes that are not intended to be executed. An Abstract Process may hide some of the required concrete operational details.

BPEL is an **Orchestration** language, serialized in XML, which specifies an executable process that involves message exchanges with other systems. This messaging facility depends on the use of the **Web Services Description Language (WSDL) 1.1** to describe outgoing and incoming messages.

BPEL in Enterprise Architect:

Enterprise Architect supports generating **BPEL** from executable processes in the **Business and Software Engineering** and the **Ultimate** editions. Although there is no standard graphical notation for **WS-BPEL**, Enterprise Architect uses **BPMN** profile as a graphical front-end to capture **BPEL** Process descriptions. With the help of the **BPMN** Profile, Enterprise Architect enables you to develop **BPEL** diagrams quickly and simply.

While **BPMN** provides a graphical notation for visualizing business process, **BPEL** provides a way for visualizing this graphical business process in XML.

Enterprise Architect supports generating:

- BPEL 1.1 [\[1238\]](#) from models created using constructs in BPMN 1.1 Profile
- BPEL 2.0 [\[1254\]](#) from models created using constructs in BPMN 2.0 Profile

Notes:

- BPEL is supported in the **Business and Software Engineering** and the **Ultimate** editions of Enterprise Architect.
- Enable BPMN 1.1 Technology for BPEL 1.1 and BPMN 2.0 Technology for BPEL 2.0 modeling in the MDG Technologies dialog (**Settings | MDG Technologies**).

Learn More:

- [BPEL](#) (Online Resource)
- [Orchestration](#) (Online Resource)
- [BPEL 1.1 Model](#) [\[1236\]](#)
- [BPEL 2.0 Model](#) [\[1254\]](#)
- [BPMN Model](#) [\[1219\]](#)

8.3.5.1 BPEL 1.1 Model

Enterprise Architect uses **BPMN 1.1** as a graphical front-end to capture **BPEL 1.1** Process description (as **BPEL 1.1** does not have not a standard graphical notation). Enterprise Architect uses the partial mapping specified in the **OMG BPMN 1.1** specification to map **BPMN 1.1** constructs to **BPEL 1.1**.

How To:

Step	Action	See Also										
1	<p>Create a BPEL 1.1 Package Structure</p> <p>Alternative:</p> <table border="1"> <thead> <tr> <th>Step</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>Create a package in the Project Browser</td> </tr> <tr> <td>b</td> <td>Click on the New Diagram icon in the Project Browser</td> </tr> <tr> <td>c</td> <td>Select BPMN 1.1 in the Select From field and BPEL in the Diagram</td> </tr> <tr> <td>d</td> <td>Click on the BPEL Process icon from the BPMN 1.1 Core toolbox at the created BPEL diagram</td> </tr> </tbody> </table>	Step	Action	a	Create a package in the Project Browser	b	Click on the New Diagram icon in the Project Browser	c	Select BPMN 1.1 in the Select From field and BPEL in the Diagram	d	Click on the BPEL Process icon from the BPMN 1.1 Core toolbox at the created BPEL diagram	<p>BPEL 1.1 Package Structure [1238]</p> <p>Project Browser Toolbar [452]</p>
Step	Action											
a	Create a package in the Project Browser											
b	Click on the New Diagram icon in the Project Browser											
c	Select BPMN 1.1 in the Select From field and BPEL in the Diagram											
d	Click on the BPEL Process icon from the BPMN 1.1 Core toolbox at the created BPEL diagram											
2	Open the BPEL diagram under the <i>BPELProcess</i> stereotyped element											
3	<p>Model the <i>BPEL Process</i> using the constructs in the BPMN 1.1 Core toolbox. Use the following constructs in this toolbox to model BPEL 1.1:</p> <p>BPMN 1.1 - Core page:</p> <ul style="list-style-type: none"> • BPEL Process • Activity • Start Event • Intermediate Event 	<p>BPMN 1.1 Core Toolbox [1220]</p> <p>Model BPEL 1.1 Process [1246]</p>										

	<ul style="list-style-type: none"> • End Event • Gateway • Pool <p>BPMN 1.1 - Relationships page:</p> <ul style="list-style-type: none"> • Sequence Flow • Message Flow • Association <p>BPMN 1.1 Types page:</p> <ul style="list-style-type: none"> • Assignment • Condition • Message • Participant • Web Service 	
4	Create Web Service for the <i>BPEL Process</i> and other related <i>Pools</i> , if required	BPEL Web Service <small>[125]</small>
5	Generate BPEL 1.1 code from the <i>BPEL Process</i>	Generate BPEL 1.1 <small>[125]</small>

Notes:

- Generating **BPEL 1.1** from a **BPMN 1.1** model is supported in the **Business and Software Engineering** and the **Ultimate** editions of Enterprise Architect.
- Ensure that **BPMN 1.1** Technology is enabled in the **MDG Technologies** dialog (**Settings | MDG Technologies**).

BPEL 1.1 Example:

- The Enterprise Architect Example file (**EAExample.EAP**) has a sample **BPMN 1.1** model from which **BPEL 1.1** can be generated.
- If you have installed Enterprise Architect at the default location, open this file: **C:\Program Files\Sparx Systems\EA\EAExample.EAP**
- The **BPMN 1.1** model package is in: **Project Models | Analysis and Business Modeling | BPEL Example | BPEL 1.1 Model**.

Modeling Restrictions:

1. Every *BPEL Process* and *Sub-Process* should start with a *Start Event* and end with an *End Event*.
 - A *Start Event* or an *End Event* should not be attached to the boundary of a *Sub-Process*.
2. *Sequence Flow* Looping is not supported - only *Activity* looping is supported. All *Sequence Flows* should flow downstream and not upstream.
3. Mapping of an *Intermediate Event* with multiple triggers to **BPEL 1.1** is not supported.
4. Mapping of *Multi-Instance Parallel While* loops to **BPEL 1.1** is not supported.
5. Mapping of *Independent Sub-Process* to **BPEL 1.1** is not supported.
6. *Pools* are treated as *black box* (i.e., they do not expose any details) and hence they cannot contain any child elements or have any incoming/outgoing *Sequence Flow* connectors.

8.3.5.1.1 Create BPEL 1.1 Model Structure

A **BPEL 1.1** model consists of a *BPEL Process* (containing a **BPEL** diagram and mappable **BPMN 1.1** constructs) and other supporting elements (like **BPMN 1.1 Assignment**, **BPMN 1.1 Web Service**) required for generating **BPEL 1.1** code. A sample **BPEL 1.1** package structure can be created in the **Project Browser**, using the **Select Model(s)** (Model Wizard) dialog. You can use this package structure as a template for developing your *BPEL Process*.

Access:

- Right-click on a package in the **Project Browser** and select the following context menu : **Add | Add a New Model using Wizard** (or)
- Select the **New Model from Pattern** icon in the **Project Browser** toolbar

Use to:

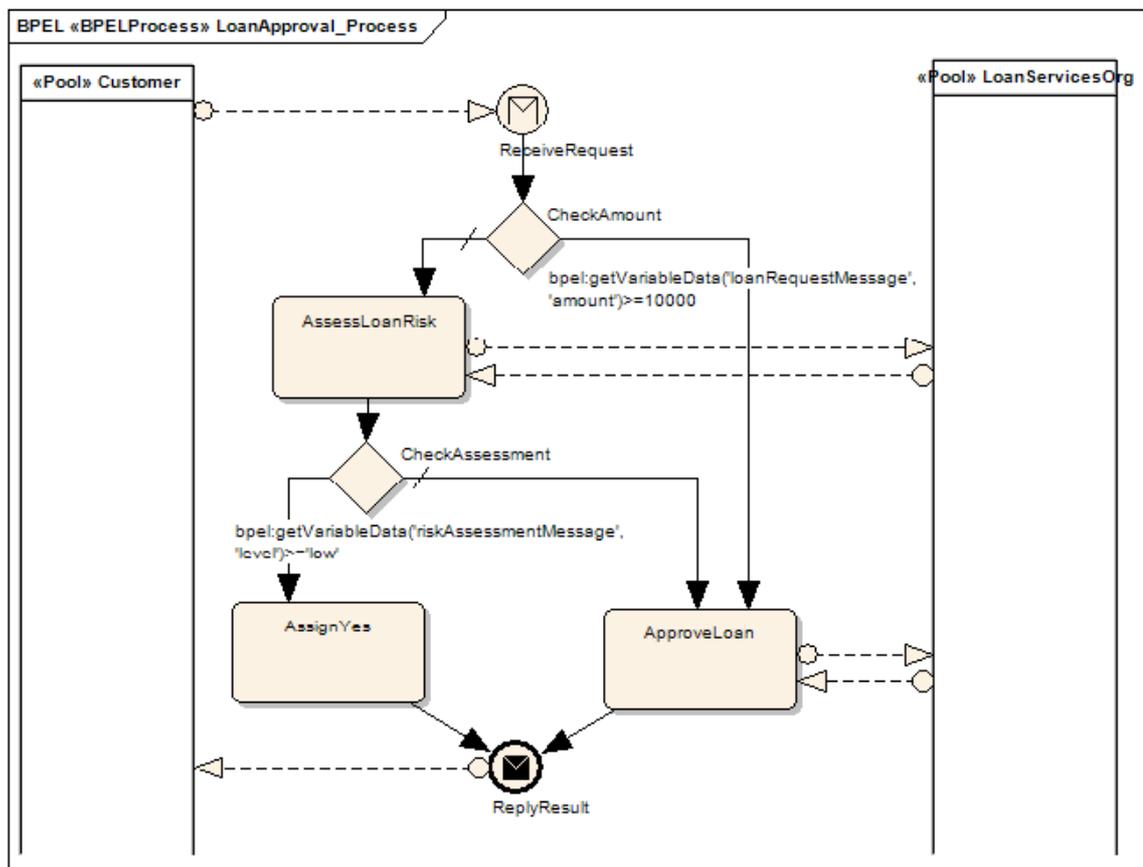
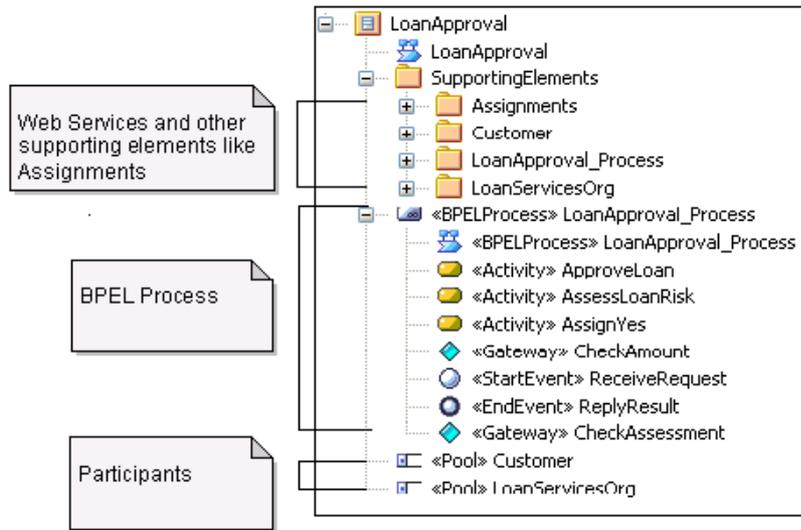
- Create a new **BPEL 1.1** package structure that can be used as a starting point for developing your *BPEL Process*.

How to:

Step	Action	See also
1	Select the root node or a package in the Project Browser	
2	Click on the New Model from Pattern icon in the Project Browser	Project Browser Toolbar 
3	Select the value BPMN 1.1 in the Technology section	Model Wizard 
4	Check the option BPEL 1.1 Model in the Name section	
5	Click on the OK button to create the sample BPEL 1.1 package structure	

Example BPEL 1.1 Package Structure:

The *BPEL Process* **LoanApproval_Process** acts as container for the **BPEL** diagram and elements. The *SupportingElements* package contains supporting components like **Assignments** and **Web Services**.



Learn More:

- [Model a BPEL 1.1 Process](#) ^[1240]
- [Create a BPEL 1.1 Web Service](#) ^[1251]
- [Generate BPEL 1.1](#) ^[1253]

8.3.5.1.2 Model a BPEL 1.1 Process

The *BPEL Process* in Enterprise Architect represents the top-level container for the BPMN 1.1 elements, from which BPEL 1.1 can be generated. Conceptually it maps to the BPEL *process* element.

BPEL Properties:

- Double-click on the *BPEL Process* in the **BPEL** diagram
- Right-click on the *BPEL Process* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>BPEL Process</i>	
Query Language	Specify the value of the language used in the <i>BPEL Process</i> for the selection of nodes in <i>Assignments</i> . Defaults to XPath 1.0	
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties [660]

Learn More:

The *BPEL Process* element acts as a container for the following **BPMN 1.1** constructs:

- [Start Event](#) [1240]
- [Intermediate Event](#) [1242]
- [Activity](#) [1244]
- [Gateway](#) [1246]
- [End Event](#) [1247]
- [Sequence Flow](#) [1248]

The following **BPMN 1.1** constructs can be optionally used in **BPEL** modeling :

- [Pool](#) [1249]
- [Assignment](#) [1250]

8.3.5.1.2.1 Start Event

A **BPMN 1.1 Start Event** indicates where a particular *Process* begins. Every *Process* in Enterprise Architect must begin with a *Start Event*.

BPEL Properties:

- Double-click on the *Start Event* in the **BPEL** diagram (or)
- Right-click on the *Start Event* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL Properties**

Reference:

Field/Button	Usage	See Also																						
Name	Specify the name for the <i>Start Event</i>																							
Trigger Type	Select a trigger for the <i>Start Event</i> , namely: <ul style="list-style-type: none"> • Conditional • Link • Message • Multiple • None • Signal • Timer 																							
Details	Depending on the selected Trigger Type , the Details tab changes as follows : <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Conditional Trigger Type</td> </tr> <tr> <td>Condition</td> <td>Select a BPMN 1.1 Condition from the list of (s) created in the SupportingElements package</td> </tr> <tr> <td colspan="2">Message Trigger Type</td> </tr> <tr> <td>Web Service</td> <td>Select a BPMN 1.1 Web Service from the package (the SupportingElements package) that represents one of the involved <i>Participants</i></td> </tr> <tr> <td>Message</td> <td>Select a BPMN 1.1 Message from the list of (s) in the selected Web Service</td> </tr> <tr> <td colspan="2">Multiple Trigger Type</td> </tr> <tr> <td>Events</td> <td>Select additional BPMN 1.1 Start Events from the list of (s) in the selected Web Service that might trigger the <i>Start Event</i> by clicking the (...) button</td> </tr> <tr> <td colspan="2">Timer Trigger Type</td> </tr> <tr> <td>Time Cycle</td> <td>Specify the value for the time duration</td> </tr> <tr> <td>Time Date</td> <td>Specify the value for the time date</td> </tr> </tbody> </table>	Field	Usage	Conditional Trigger Type		Condition	Select a BPMN 1.1 Condition from the list of (s) created in the SupportingElements package	Message Trigger Type		Web Service	Select a BPMN 1.1 Web Service from the package (the SupportingElements package) that represents one of the involved <i>Participants</i>	Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service	Multiple Trigger Type		Events	Select additional BPMN 1.1 Start Events from the list of (s) in the selected Web Service that might trigger the <i>Start Event</i> by clicking the (...) button	Timer Trigger Type		Time Cycle	Specify the value for the time duration	Time Date	Specify the value for the time date	
Field	Usage																							
Conditional Trigger Type																								
Condition	Select a BPMN 1.1 Condition from the list of (s) created in the SupportingElements package																							
Message Trigger Type																								
Web Service	Select a BPMN 1.1 Web Service from the package (the SupportingElements package) that represents one of the involved <i>Participants</i>																							
Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service																							
Multiple Trigger Type																								
Events	Select additional BPMN 1.1 Start Events from the list of (s) in the selected Web Service that might trigger the <i>Start Event</i> by clicking the (...) button																							
Timer Trigger Type																								
Time Cycle	Specify the value for the time duration																							
Time Date	Specify the value for the time date																							
Assignments	Select one or more <i>Assignment</i> elements created in the SupportingElements package in this tab (Optional)	Assignment ^[1250] Create BPEL 1.1 Model Structure <small>[1238]</small>																						
OK	Save the values entered in the dialog																							

Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties ^[660]

Notes:

- Either set **Time Cycle** or **Time Date** for **Timer Trigger Type** but not both as they are mutually exclusive fields.
- **Link, None** and **Signal Event** types cannot be mapped to **BPEL 1.1**

8.3.5.1.2.2 Intermediate Event

A **BPMN 1.1 Intermediate Event** indicates where an event occurs somewhere between the start and end of a *Process*.

BPEL Properties:

- Double-click on the *Intermediate Event* in the **BPEL** diagram
- Right-click on the *Intermediate Event* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL Properties**

Reference:

Field/Button	Usage	See Also										
Name	Specify the name for the <i>Intermediate Event</i>											
Trigger Type	Select the type for the <i>Intermediate Event</i> , namely: <ul style="list-style-type: none"> • Cancel • Compensation • Conditional • Error • Link • Message • Multiple • None • Signal • Timer 											
Details	Depending on the selected Trigger Type , the Details tab changes as follows : <table border="1" data-bbox="491 1666 1209 2045"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Compensation Trigger Type</td> </tr> <tr> <td>Activity</td> <td>Select an BPMN 1.1 Task from the list of all Tasks in the <i>Process</i></td> </tr> <tr> <td colspan="2">Conditional Trigger Type</td> </tr> <tr> <td>Condition</td> <td>Select a BPMN 1.1 Condition from the list of all Conditions in the <i>Process</i></td> </tr> </tbody> </table>	Field	Usage	Compensation Trigger Type		Activity	Select an BPMN 1.1 Task from the list of all Tasks in the <i>Process</i>	Conditional Trigger Type		Condition	Select a BPMN 1.1 Condition from the list of all Conditions in the <i>Process</i>	Create BPEL 1.1 Model Structure ^[1238]
Field	Usage											
Compensation Trigger Type												
Activity	Select an BPMN 1.1 Task from the list of all Tasks in the <i>Process</i>											
Conditional Trigger Type												
Condition	Select a BPMN 1.1 Condition from the list of all Conditions in the <i>Process</i>											

		Condition(s) created in the SupportingElements package	
	Error Trigger Type		
Error Code		Specify the required error code	
	Link Trigger Type		
Link Event		Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i>) from the list of <i>Link Event</i> elements under the current <i>Process / Sub-Process</i>	
	Message Trigger Type		
Web Service		Select a BPMN 1.1 Web Service from the list of BPMN 1.1 Web Services (in the SupportingElements package) that represents one of the involved <i>Participants</i>	
Message		Select a BPMN 1.1 Message from the list of BPMN 1.1 Messages (s) in the selected Web Service	
	Multiple Trigger Type		
Events		Select additional BPMN 1.1 Intermediate Event from the list of events in this <i>Process</i> by clicking the Events button	
	Timer Trigger Type		
Time Cycle		Specify the value for the time duration	
Time Date		Specify the value for the time date	
OK	Save the values entered in the dialog		
Cancel	Discard the values entered in the dialog		
Help	Display this Help topic		
General	Open the UML Properties dialog		Properties 

Notes:

- **Cancel**, **None** and **Signal Event** types cannot be mapped to **BPEL 1.1**.
- Either set **Time Cycle** or **Time Date** for **Timer Trigger Type** but not both as they are mutually exclusive fields.
- For a **Compensation Intermediate Event** edge-mounted on an *Activity*, create a **BPMN 1.1 Association** from this event to the **Compensation Activity**. Ensure that the *IsCompensation* tag for the *Activity* is set to true.
- **Link Intermediate Event** can be used either as a GOTO or an off-page connector. So, this event can have either incoming or outgoing *Sequence Flows* - but not both.

8.3.5.1.2.3 Activity

A **BPMN 1.1 Activity** represents work that is performed within a *Process*. An *Activity* can be modeled as a:

- **Sub-Process** - a compound *Activity* that is defined as a flow of other **BPMN 2.0** elements or
- **Task** - an atomic *Activity* that cannot be broken down into a smaller unit.

Activities - both *Tasks* and *Sub-Processes* - can also, optionally, act as *Looping constructs*. The **OMG BPMN 2.0 Specification** defines two types of *Looping construct*:

- **Standard Loop** (*while* or *until*)
- **Multi-Instance Loop** (*for each*)

BPEL Properties:

- Double-click on the *Activity* in the **BPEL** diagram
- Right-click on the *Activity* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also						
Name	Specify the name for the <i>Activity</i>							
Type	Specify whether the <i>Activity</i> is a : <ul style="list-style-type: none"> • Task (or) • Sub-Process 							
Task Type / SubProcess Type	Depending on the value selected in the Type field, the Task Type / SubProcess Type field has the following values : <p>Task Type :</p> <ul style="list-style-type: none"> • Manual • None • Receive • Reference • Script • Send • Service • User <p>SubProcess Type :</p> <ul style="list-style-type: none"> • Embedded • References • Reusable 							
Details	Depending on the selected Task Type , the Details tab changes as follows : <table border="1" data-bbox="491 1776 1214 2000"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Receive Task Type</td> </tr> <tr> <td>Web Service</td> <td>Select a BPMN 1.1 Web Service from the p (in the SupportingElements package) that represents one of the involved <i>Participants</i></td> </tr> </tbody> </table>	Field	Usage	Receive Task Type		Web Service	Select a BPMN 1.1 Web Service from the p (in the SupportingElements package) that represents one of the involved <i>Participants</i>	
Field	Usage							
Receive Task Type								
Web Service	Select a BPMN 1.1 Web Service from the p (in the SupportingElements package) that represents one of the involved <i>Participants</i>							

	<table border="1"> <tr> <td data-bbox="494 253 738 338">Message</td> <td data-bbox="738 253 1206 338">Select a BPMN 1.1 Message from the list of (s) in the selected Web Service</td> </tr> <tr> <td data-bbox="494 338 738 423">Instantiate</td> <td data-bbox="738 338 1206 423">Select <i>True</i> if this is the first <i>Activity</i> after the <i>Event</i>^[1240], otherwise select <i>False</i>.</td> </tr> <tr> <td colspan="2" data-bbox="494 423 1206 477"></td> </tr> <tr> <td colspan="2" data-bbox="494 477 1206 530">Reference Task Type</td> </tr> <tr> <td data-bbox="494 530 738 616">Activity</td> <td data-bbox="738 530 1206 616">Select an BPMN 1.1 Task from the list of all in the <i>Process</i></td> </tr> <tr> <td colspan="2" data-bbox="494 616 1206 669"></td> </tr> <tr> <td colspan="2" data-bbox="494 669 1206 723">Send Task Type</td> </tr> <tr> <td data-bbox="494 723 738 831">Web Service</td> <td data-bbox="738 723 1206 831">Select a BPMN 1.1 Web Service from the package (in the SupportingElements package) that represents one of the involved <i>Participants</i></td> </tr> <tr> <td data-bbox="494 831 738 916">Message</td> <td data-bbox="738 831 1206 916">Select a BPMN 1.1 Message from the list of (s) in the selected Web Service</td> </tr> <tr> <td colspan="2" data-bbox="494 916 1206 969"></td> </tr> <tr> <td colspan="2" data-bbox="494 969 1206 1023">Service / User Task Type</td> </tr> <tr> <td data-bbox="494 1023 738 1131">Web Service</td> <td data-bbox="738 1023 1206 1131">Select a BPMN 1.1 Web Service from the package (in the SupportingElements package) that represents one of the involved <i>Participants</i></td> </tr> <tr> <td data-bbox="494 1131 738 1238">Input Message</td> <td data-bbox="738 1131 1206 1238">Select a BPMN 1.1 Message from the list of (s) in the selected Web Service that represents input element</td> </tr> <tr> <td data-bbox="494 1238 738 1346">Output Message</td> <td data-bbox="738 1238 1206 1346">Select a BPMN 1.1 Message from the list of (s) in the selected Web Service that represents output element</td> </tr> <tr> <td colspan="2" data-bbox="494 1346 1206 1400"></td> </tr> <tr> <td colspan="2" data-bbox="494 1400 1206 1489">Depending on the selected SubProcess Type, the Details tab changes as follows :</td> </tr> <tr> <td data-bbox="494 1489 738 1565">Field</td> <td data-bbox="738 1489 1206 1565">Usage</td> </tr> <tr> <td colspan="2" data-bbox="494 1565 1206 1619">References SubProcess Type</td> </tr> <tr> <td data-bbox="494 1619 738 1695">SubProcess</td> <td data-bbox="738 1619 1206 1695">Select an BPMN 1.1 Sub-Process from the <i>Activities</i> in the <i>Process</i></td> </tr> <tr> <td colspan="2" data-bbox="494 1695 1206 1704"></td> </tr> </table>	Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service	Instantiate	Select <i>True</i> if this is the first <i>Activity</i> after the <i>Event</i> ^[1240] , otherwise select <i>False</i> .			Reference Task Type		Activity	Select an BPMN 1.1 Task from the list of all in the <i>Process</i>			Send Task Type		Web Service	Select a BPMN 1.1 Web Service from the package (in the SupportingElements package) that represents one of the involved <i>Participants</i>	Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service			Service / User Task Type		Web Service	Select a BPMN 1.1 Web Service from the package (in the SupportingElements package) that represents one of the involved <i>Participants</i>	Input Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service that represents input element	Output Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service that represents output element			Depending on the selected SubProcess Type , the Details tab changes as follows :		Field	Usage	References SubProcess Type		SubProcess	Select an BPMN 1.1 Sub-Process from the <i>Activities</i> in the <i>Process</i>			
Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service																																									
Instantiate	Select <i>True</i> if this is the first <i>Activity</i> after the <i>Event</i> ^[1240] , otherwise select <i>False</i> .																																									
Reference Task Type																																										
Activity	Select an BPMN 1.1 Task from the list of all in the <i>Process</i>																																									
Send Task Type																																										
Web Service	Select a BPMN 1.1 Web Service from the package (in the SupportingElements package) that represents one of the involved <i>Participants</i>																																									
Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service																																									
Service / User Task Type																																										
Web Service	Select a BPMN 1.1 Web Service from the package (in the SupportingElements package) that represents one of the involved <i>Participants</i>																																									
Input Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service that represents input element																																									
Output Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service that represents output element																																									
Depending on the selected SubProcess Type , the Details tab changes as follows :																																										
Field	Usage																																									
References SubProcess Type																																										
SubProcess	Select an BPMN 1.1 Sub-Process from the <i>Activities</i> in the <i>Process</i>																																									
Assignments	Select one or more <i>Assignment</i> elements created in the SupportingElements package in this tab (Optional)	Assignment ^[1250] Create BPEL 1.1 Model Structure ^[1238]																																								
Loop Details	Activities can be repeated sequentially, behaving like a loop. Specify the <i>Activity</i> looping details in this tab (Optional)																																									
	<table border="1"> <tr> <td data-bbox="494 1960 738 2016">Field</td> <td data-bbox="738 1960 1206 2016">Usage</td> </tr> </table>	Field	Usage																																							
Field	Usage																																									

	<table border="1"> <tr> <td>Type</td> <td>Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop • Multinstance for creating a <i>for each</i> loop </td> </tr> <tr> <td>Condition</td> <td>Enter a boolean condition for Standard loop numeric condition for Multinstance loop</td> </tr> <tr> <td>Min Value</td> <td>Specify the minimum value for the loop evaluation a Standard loop (Optional)</td> </tr> <tr> <td>Max Value</td> <td>Specify the maximum value for the loop evaluation a Standard loop (Optional)</td> </tr> <tr> <td>Test Time</td> <td>Select After to define a <i>while</i> Standard loop Before to define an <i>until</i> Standard loop</td> </tr> </table>	Type	Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop • Multinstance for creating a <i>for each</i> loop 	Condition	Enter a boolean condition for Standard loop numeric condition for Multinstance loop	Min Value	Specify the minimum value for the loop evaluation a Standard loop (Optional)	Max Value	Specify the maximum value for the loop evaluation a Standard loop (Optional)	Test Time	Select After to define a <i>while</i> Standard loop Before to define an <i>until</i> Standard loop	
Type	Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop • Multinstance for creating a <i>for each</i> loop 											
Condition	Enter a boolean condition for Standard loop numeric condition for Multinstance loop											
Min Value	Specify the minimum value for the loop evaluation a Standard loop (Optional)											
Max Value	Specify the maximum value for the loop evaluation a Standard loop (Optional)											
Test Time	Select After to define a <i>while</i> Standard loop Before to define an <i>until</i> Standard loop											
OK	Save the values entered in the dialog											
Cancel	Discard the values entered in the dialog											
Help	Display this Help topic											
General	Open the UML Properties dialog	Properties 										

Notes:

- **Manual** and **Script** types cannot be mapped to **BPEL 1.1**
- **Reusable Sub-Process** types cannot be mapped to **BPEL 1.1**

8.3.5.1.2.4 Gateway

A **BPMN 1.1 Gateway** control the way in which *Sequence Flows* converge and diverge within a *Process*. They provide a gating mechanism that either allows or blocks a *Sequence Flow*.

BPEL Properties:

- Double-click on the *Gateway* in the **BPEL** diagram
- Right-click on the *Gateway* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL Properties**

Reference:

Field/Button	Usage	See Also				
Name	Specify the name for the <i>Gateway</i>					
Gateway	Select the type for the <i>Gateway</i> , namely: <ul style="list-style-type: none"> • Complex • Exclusive • Inclusive • Parallel 					
Details	Depending on the selected Gateway , the Details tab changes as follows : <table border="1" data-bbox="512 1935 1295 2045"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Field	Usage			
Field	Usage					

	<table border="1"> <tr> <td colspan="2">Exclusive Gateway</td> </tr> <tr> <td>Exclusive Type</td> <td> <p>Data - allows alternate paths are taken based on evaluation of conditions</p> <p>Event - allows alternate paths are taken based on the occurrence of events</p> </td> </tr> </table>	Exclusive Gateway		Exclusive Type	<p>Data - allows alternate paths are taken based on evaluation of conditions</p> <p>Event - allows alternate paths are taken based on the occurrence of events</p>	
Exclusive Gateway						
Exclusive Type	<p>Data - allows alternate paths are taken based on evaluation of conditions</p> <p>Event - allows alternate paths are taken based on the occurrence of events</p>					
OK	Save the values entered in the dialog					
Cancel	Discard the values entered in the dialog					
Help	Display this Help topic					
General	Open the UML Properties dialog	Properties 6601				

Notes:

- The target of the outgoing *Sequence Flows* of this **Event Exclusive Gateway** must be either a:
 - **Receive Task**
 - **Message** or **Timer Intermediate Event**
- If a **Message Task** is one of the targets of the outgoing *Sequence Flow* of an **Event Exclusive Gateway**, then a **Message Intermediate Event** cannot be the target of the other outgoing *Sequence Flows* of this **Event Exclusive Gateway**

8.3.5.1.2.5 End Event

A **BPMN 1.1 End Event** indicates where a particular *Process* ends. Every *Process* in Enterprise Architect must end with an End Event.

BPEL Properties:

- Double-click on the *End Event* in the **BPEL** diagram
- Right-click on the *End Event* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>End Event</i>	
Result Type	Select the type for the <i>End Event</i> , namely: <ul style="list-style-type: none"> • Cancel • Compensation • Error • Link • Message • Multiple • None • Signal • Terminate 	
Details	Depending on the selected Result Type , the Details tab changes as follows :	

	Field	Usage	
	Compensation Result Type		
	Activity	Select an BPMN 1.1 Task from the list of all in the <i>Process</i>	
	Error Trigger Type		
	Error Code	Specify the required error code	
	Message Result Type		
	Web Service	Select a BPMN 1.1 Web Service from the p (the <i>SupportingElements</i> package) that rep one of the involved <i>Participants</i>	
	Message	Select a BPMN 1.1 Message from the list of (s) in the selected Web Service	
	Multiple Result Type		
	Events	Select additional BPMN 1.1 Start Events fro events in this <i>Process</i> that might trigger the clicking the (...) button	
Assignments	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional)		Assignment ^[1250] Create BPEL 1.1 Model Structure ^[1238]
OK	Save the values entered in the dialog		
Cancel	Discard the values entered in the dialog		
Help	Display this Help topic		
General	Open the UML Properties dialog		Properties ^[660]

Notes:

- **Cancel**, **Link**, **None** and **Signal Event** types cannot be mapped to **BPEL 1.1**

8.3.5.1.2.6 Sequence Flow

A **BPMN 1.1 Sequence Flow** connector shows the order in which the activities (*Tasks* and *Events*) are performed in a *BPEL Process*.

BPEL Properties:

- Double-click on the *Sequence Flow* in the **BPEL** diagram
- Right-click on the *Sequence Flow* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL Properties**

Reference:

Field/Button	Usage	See Also
Condition Type	Specify the type of the condition on the <i>Sequence Flow</i> , namely : <ul style="list-style-type: none"> • None • Default • Expression 	
Expression	This field gets enabled when the Condition Type is set to Expression . Specify a boolean expression to act as a gating condition.	
Ordering	This field gets enabled when the Condition Type is set to Expression . Specify a numerical value that determines the order in which the condition set in the Expression field is to be evaluated.	
Assignments	Select one or more <i>Assignment</i> elements created in the SupportingElements package in this tab (Optional)	Assignment ^[1250] Create BPEL 1.1 Model Structure ^[1238]
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties ^[660]

8.3.5.1.2.7 Pool

A **BPMN 1.1 Pool** represents a **Participant** in a *Process* and does not map to any specific **BPEL 1.1** element. Enterprise Architect uses *Pools* to represent external **Participants**, with which the *BPEL Process* communicates. These are 'black box' *pools* i.e., they are abstract and do not expose any details (they do not contain any **BPMN 1.1** elements inside them).

BPEL Properties:

- Double-click on the *Pool* in the **BPEL** diagram
- Right-click on the *Pool* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>Pool</i>	
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	

General	Open the UML Properties dialog	Properties [660]
---------	---------------------------------------	-------------------------------------

Notes:

- A *BPEL Process* should not contain a *Pool* as its child element.
- A **BPEL** diagram under a *BPEL Process* contains an implicit *Pool* - so it is invalid to add a *Pool* in this **BPEL** diagram to represent the *BPEL Process*.
- *Pool* cannot have any incoming or outgoing *Sequence Flow* connectors - it can have only incoming or outgoing *Message Flow* connectors.

8.3.5.1.2.8 Assignment

A **BPMN 1.1 Assignment** element enables data to be copied between messages and new data to be inserted, using expressions within a *BPEL Process*. A **BPMN 1.1 Assignment** element maps to a **BPEL 1.1 assign** activity and copies the specified value from the source to the target.

In Enterprise Architect, *Assignment* elements should be created in the **Assignments** package in **SupportingElements** [1258]. If they are created elsewhere, they cannot be enacted correctly.

Access: [View](#) | [Toolbox > More tools](#) | [BPMN 1.1](#) | [BPMN 1.1 Types](#)

BPEL Properties:

- Double-click on the *Assignment* in the **BPEL** diagram
- Right-click on the *Assignment* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>Assignment</i>	
Assign Time	Select either Start or End to determine whether the assignment occurs at the start or end of an <i>Activity/Event</i> .	
Copy From section		
Type	Depending on the value selected in this field, further details are required	
Literal	Specify a literal value	
Expression / Message	Select a BPMN 1.1 Message in the package representing the <i>BPEL Process / Pool</i> under the SupportingElements package	Create BPEL 1.1 Model Structure [1238] Create a BPEL 1.1 Web Service [1257]
Part	Select a BPMN 1.1 Property belonging to the selected Message (Optional)	
Copy To section		
Message	Select a BPMN 1.1 Message in the package representing the <i>BPEL Process / Pool</i> under the SupportingElements	Create BPEL 1.1 Model Structure [1238]

	package	Create a BPEL 1.1 Web Service <small>[1251]</small>
Part	Select a BPMN 1.1 Property belonging to the selected Message (Optional)	
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties <small>[660]</small>

Notes:

- Messages are created when you create a **Web Services**
- If you select **Expression**, Enterprise Architect uses the **getVariableData XPATH 1.0** function to create the expression from the selected **Message** and **Part**.

8.3.5.1.3 Create BPEL 1.1 Web Service

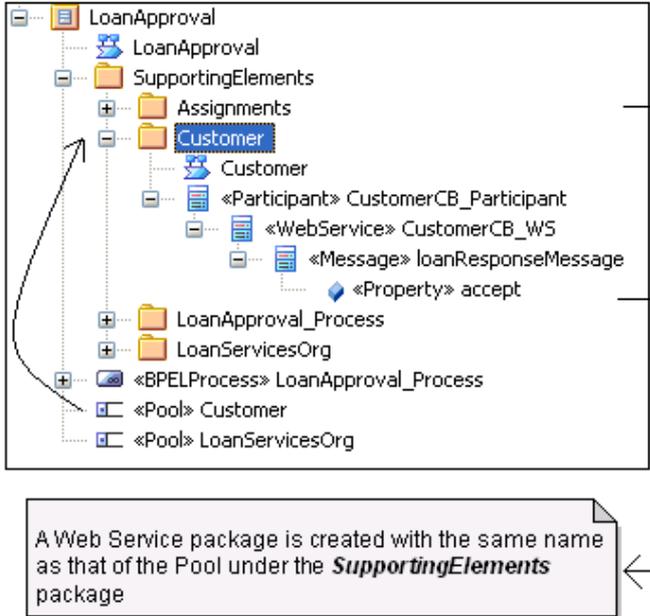
BPEL is an **Orchestration** language, which orchestrates services that are exposed using **WSDL 1.1**. It coordinates the execution of the various operations of these involved services. **BPEL 1.1** supports **WSDL 1.1 one-way** and **request-response** operations only (and not **solicit-response** and **notification** operations). Enterprise Architect enables you to create, for *BPEL Process* and *Pool* elements, **Web Service Operations** that support either *synchronous* (request-response) or *asynchronous* (one-way) interactions.

Access: Right-click on a *BPEL Process* or *Pool* and select the following context menu : **BPEL | Create WebService**

Reference:

Field/Button	Usage	See Also
Operation	Specify whether to create a new Web Service or create one from an existing WSDL created / imported into EA (using the Import WSDL dialog) by selecting either : <ul style="list-style-type: none"> • Create New (or) • Create from existing WSDL 	Model WSDL <small>[1612]</small> Import WSDL <small>[1622]</small>
Type	Specify whether to create a <i>one-way</i> or <i>request-response</i> Operation by selecting either : <ul style="list-style-type: none"> • Asynchronous (or) • Synchronous 	
Partnership Details	The interaction between the Web Service and the <i>BPEL Process</i> is modeled as a BPEL partnerLink . For this interaction, provide a:	
	Field	Usage
	Name	Name of the BPEL partnerLink
Role	Name of either <i>myRole</i> (if this operation belongs to the <i>BPEL Process</i> Web Service) or <i>partnerLinkRole</i> (if this operation belongs to the Web Service <i>Pool</i>) of the BPEL partnerLink .	

Web Service	Field	Usage
	When Create New is selected in Operation field:	
	Web Service Name	Name of the WSDL 1.1 Web Service
	PortType	Name of the WSDL 1.1 <i>PortType</i> (Interface
	Operation	Name of the WSDL 1.1 <i>PortType</i> <i>Operation</i>
	When Create from existing WSDL PortType Operation is selected in Operation field:	
	WSDL Package	Select an existing WSDL package created / imported into EA
	Web Service Name	Name of the WSDL 1.1 Web Service Default: name of the selected WSDL Packa
PortType	Select a WSDL 1.1 <i>PortType</i> from the selected WSDL Package	
Operation	Select a WSDL 1.1 <i>PortType</i> <i>Operation</i> in the selected PortType	
Input	Field	Usage
	When Create New is selected in Operation field:	
	Message Name	Name of the WSDL 1.1 <i>Message</i>
	Properties	Press (...) button to enter the WSDL 1.1 <i>Message Part Name</i> and XSD Type
	When Create from existing WSDL PortType Operation is selected in Operation field:	
The fields in this tab are pre-filled with the details of the input WSDL 1.1 <i>Message</i> (of the WSDL 1.1 <i>PortType</i> <i>Operation</i> selected in the Operation field in the Web Service tab)		
Output	Field	Usage
	When Create New is selected in Operation field:	
	Message Name	Name of the WSDL 1.1 <i>Message</i>
	Properties	Press (...) button to enter the WSDL 1.1 <i>Message Part Name</i> and XSD Type
	When Create from existing WSDL PortType Operation is selected in Operation field:	
The fields in this tab are pre-filled with the details of the output WSDL 1.1 <i>Message</i>		

	<p>Message (of the WSDL 1.1 PortType Operation selected in the Op field in the Web Service tab)</p>	
OK	<p>Create a Web Service Operation based on the values entered in the dialog.</p> <p>The Operation is created in a package, whose name is the same as that of the <i>BPEL Process / Pool</i> from from which this dialog is invoked, under the SupportingElements package</p> 	<p>SupportingElements ¹²⁵⁶</p>
Cancel	<p>Discard the values entered in the dialog and abort creating the Web Service Operation</p>	
Help	<p>Display this Help topic</p>	

Notes:

- The **Output** tab is not applicable for an **Asynchronous** operation

8.3.5.1.4 Generate BPEL 1.1

BPEL 1.1 code can be generated from a *BPEL Process*. Enterprise Architect validates the *BPEL Process* before generating the **BPEL 1.1** code. In addition to generating the **BPEL 1.1** code, **WSDL 1.1** files are generated for the *BPEL Process* and all the involved *Pools* (provided **Web Services** are defined for them).

Access: Right-click on a BPEL Process and select the following context menu: **BPEL | Generate BPEL**

Reference:

Field/Button	Usage	See Also
File Name	Specify the path where the BPEL 1.1 file is to be generated	
Namespace	Double-click on an entry (if any) in this field to open the	

Details	Namespace Details dialog and add / edit the namespace details The entry DefaultPool represents the current <i>BPEL Process</i>	
Generate BPEL	Validate the model and generate BPEL 1.1	BPEL Model Validation ^[1271]
Close	Close this dialog	
Help	Display this Help topic	
View BPEL	View the generated BPEL 1.1 file	

Learn More:

- [Model a BPEL 1.1 Process](#)^[1240]
- [Create a BPEL 1.1 Web Service](#)^[1251]

8.3.5.2 BPEL 2.0 Model

Enterprise Architect uses **BPMN 2.0** as a graphical front-end to capture **BPEL 2.0 Process** description (as **BPEL 2.0** does not have not a standard graphical notation). Enterprise Architect uses the partial mapping specified in the **BPMN 2.0** specification to map **BPMN 2.0** constructs to **BPEL 2.0**.

How To:

Step	Action	See Also										
1	<p>Create a BPEL 2.0 Package Structure</p> <p>Alternative:</p> <table border="1"> <thead> <tr> <th>Step</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>Create a package in the Project Browser</td> </tr> <tr> <td>b</td> <td>Click on the New Diagram icon in the Project Browser</td> </tr> <tr> <td>c</td> <td>Select BPMN 2.0 in the Select From field and BPEL in the Diagram</td> </tr> <tr> <td>d</td> <td>Click on the BPEL icon from the BPMN 2.0 - Business Process to into the created BPEL diagram</td> </tr> </tbody> </table>	Step	Action	a	Create a package in the Project Browser	b	Click on the New Diagram icon in the Project Browser	c	Select BPMN 2.0 in the Select From field and BPEL in the Diagram	d	Click on the BPEL icon from the BPMN 2.0 - Business Process to into the created BPEL diagram	<p>BPEL 2.0 Package Structure^[1256]</p> <p>Project Browser Toolbar^[452]</p>
Step	Action											
a	Create a package in the Project Browser											
b	Click on the New Diagram icon in the Project Browser											
c	Select BPMN 2.0 in the Select From field and BPEL in the Diagram											
d	Click on the BPEL icon from the BPMN 2.0 - Business Process to into the created BPEL diagram											
2	Open the BPEL diagram under the <i>BPELProcess</i> stereotyped element											
3	<p>Model the <i>BPEL Process</i> using the constructs in the BPMN 2.0 - Business Process toolbox. Use the following constructs in this toolbox to model BPEL 2.0:</p> <p>BPMN 2.0 - Business Process page:</p> <ul style="list-style-type: none"> • BPEL • Activity • Data Object • Start Event • Intermediate Event 	<p>BPMN 2.0 Business Process Toolbox^[1221]</p> <p>Model BPEL 2.0 Process^[1257]</p>										

	<ul style="list-style-type: none"> • Event • Gateway • Pool <p>BPMN 2.0 - Business Process Connectors page:</p> <ul style="list-style-type: none"> • Sequence Flow • Association • Message Flow <p>BPMN 2.0 Types page:</p> <ul style="list-style-type: none"> • Assignment • Error • Interface • Operation • Participant • Property 	
4	Create Web Service Operations for the <i>BPEL Process</i> and other related <i>Pools</i> , if required	Web Service Operation <small>[1268]</small>
5	Generate BPEL 2.0 code from the <i>BPEL Process</i>	Generate BPEL 2.0 <small>[1271]</small>

Notes:

- Generating **BPEL 2.0** from a **BPMN 2.0** model is supported in the **Business and Software Engineering** and the **Ultimate** editions of Enterprise Architect.
- Ensure that **BPMN 2.0** Technology is enabled in the **MDG Technologies** dialog (**Settings | MDG Technologies**).

BPEL 2.0 Example:

- The Enterprise Architect Example file (*EAExample.EAP*) has a sample **BPMN 2.0** model from which **BPEL 2.0** can be generated.
- If you have installed Enterprise Architect at the default location, open this file: **C:\Program Files\Sparx Systems\EA\EAExample.EAP**
- The **BPMN 2.0** model package is in: **Project Models | Analysis and Business Modeling | BPEL Example | BPEL 2.0 Model**.

Modeling Restrictions:

1. Every *BPEL Process* and *Sub-Process* should start with a *Start Event* and end with an *End Event*.
 - *Boundary Start* and *End Events* are not supported
2. *Sequence Flow* Looping is not supported in both **Normal** and **Exception Paths**.
 - All *Sequence Flows* should flow downstream and not upstream.
3. *Sub-Process* cannot be a **Loop** node and have boundary *Intermediate Events*.
4. *Event Sub-Process* cannot act as a **Loop** Node.
5. *Assignments* are not supported on :
 - *Start Events* on *Event Sub-Process*
 - *End Events*
 - *Sub-Process*
 - *Boundary Intermediate Event*
 - *Gateway*
 - *Task* and *Intermediate Event* that immediately follow an *XOR Event Gateway*
6. **Exception Path** has to merge back into the **Normal Path**.
 - An exception to this rule is boundary *Compensation Intermediate Event* which should have a **BPMN 2.0 Association** to a *Compensation Activity* (that has no incoming or outgoing *Sequence Flows*)

- Multiple **Exception Paths** from an *Activity* must join at the same location in the **Normal Path**
 - An **Exception Path** should not cross another **Exception Path**.
7. *Activities* in an **Exception Path** cannot have boundary **Intermediate Events**.
8. *Pools* are treated as *black boxes* (i.e., they do not expose any details) and hence they cannot contain any child elements or have any incoming/outgoing *Sequence Flow* connectors.

8.3.5.2.1 Create BPEL 2.0 Model Structure

A **BPEL 2.0** model consists of a *BPEL Process* (containing a **BPEL** diagram and mappable **BPMN 2.0** constructs) and other supporting elements (like **BPMN 2.0 Assignment**, **BPMN 2.0 Operation**) required for generating a **BPEL 2.0** code. A sample **BPEL 2.0** package structure can be created in the **Project Browser**, using the **Select Model(s)** (Model Wizard) dialog. You can use this package structure as a template for developing your *BPEL Process*.

Access:

- Right-click on a package in the **Project Browser** and select the following context menu : **Add | Add a New Model using Wizard** (or)
- Select the **New Model from Pattern** icon in the **Project Browser** toolbar

Use to:

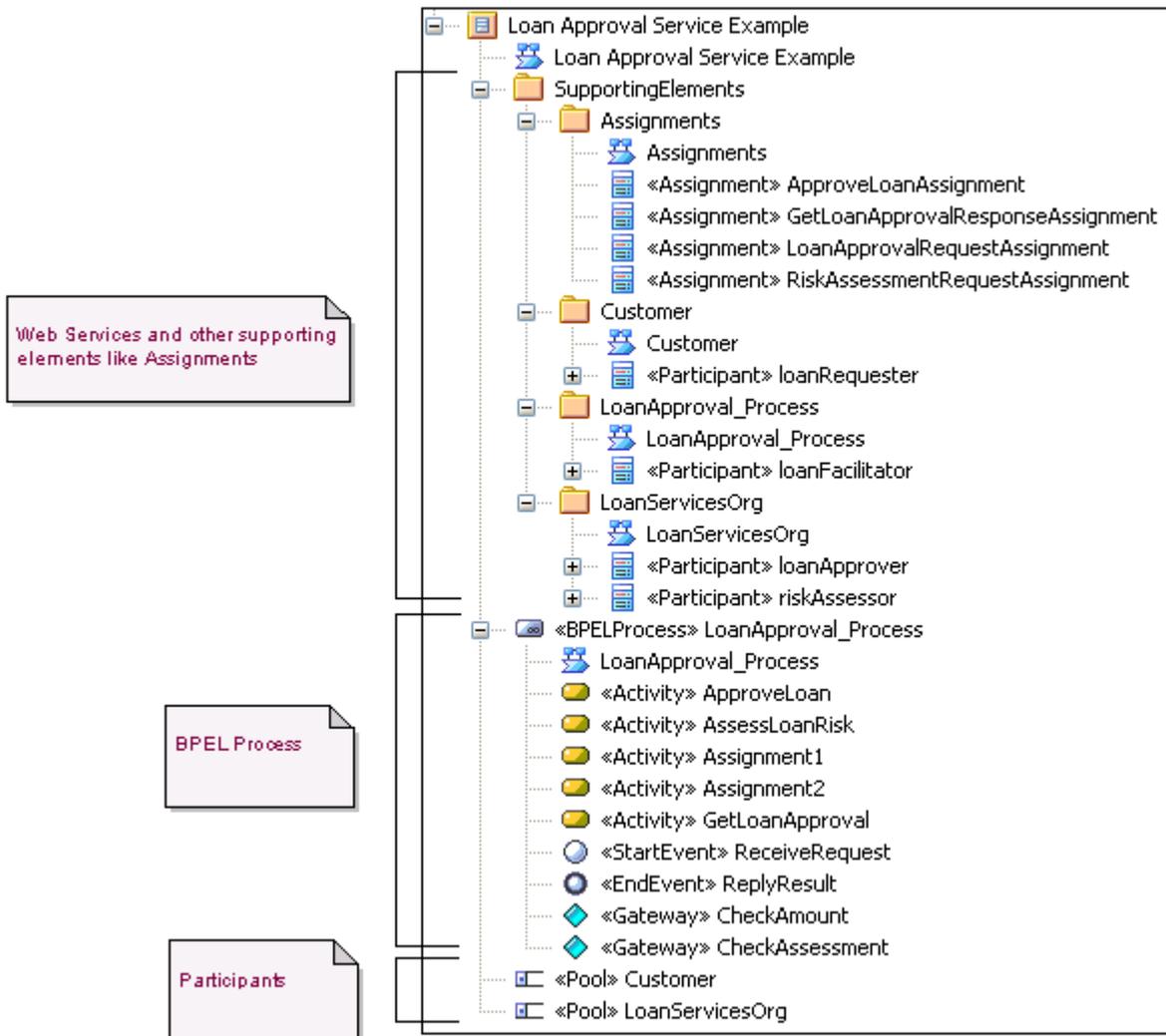
- Create a new **BPEL 2.0** package structure that can be used as a starting point for developing your *BPEL Process*.

How to:

Step	Action	See also
1	Select the root node or a package in the Project Browser	
2	Click on the New Model from Pattern icon in the Project Browser	Project Browser Toolbar [452]
3	Select the value BPMN 2.0 in the Technology section	Model Wizard [518]
4	Check the option BPEL 2.0 Model in the Name section	
5	Click on the OK button to create the sample BPEL 2.0 package structure	

Example BPEL 2.0 Package Structure:

The *BPEL Process* **LoanApproval_Process** acts as container for the **BPEL** diagram and elements. The **SupportingElements** package contains supporting components like **Assignments** and **Web Service Operations**.

**Learn More:**

- [Model a BPEL 2.0 Process](#)^[1257]
- [Create a Web Service Operation](#)^[1268]
- [Generate BPEL 2.0](#)^[1271]

8.3.5.2.2 Model a BPEL 2.0 Process

The *BPEL Process* in Enterprise Architect represents the top-level container for the **BPMN 2.0** elements, from which **BPEL 2.0** can be generated. Conceptually it maps to the BPEL *process* element.

BPEL Properties:

- Double-click on the *BPEL Process* in the **BPEL** diagram (or)
- Right-click on the *BPEL Process* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>BPEL Process</i>	
Query Language	Specify the value of the language used in the <i>BPEL Process</i> for the selection of nodes in <i>Assignments</i> . Defaults to XPath 1.0	
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties [660]

Learn More:

The *BPEL Process* element acts as a container for the following **BPMN 2.0** constructs:

- [Start Event](#) [1258]
- [Intermediate Event](#) [1259]
- [Activity](#) [1261]
- [Gateway](#) [1263]
- [End Event](#) [1264]
- [Data Object](#) [1265]
- [Property](#) [1266]
- [Sequence Flow](#) [1266]

The following BPMN 2.0 constructs can be optionally used in BPEL modeling:

- [Pool](#) [1249]
- [Assignment](#) [1250]

8.3.5.2.2.1 Start Event

A **BPMN 2.0 Start Event** indicates where a particular *Process* begins. Every *Process* in Enterprise Architect must begin with a *Start Event*.

BPEL Properties:

- Double-click on the *Start Event* in the **BPEL** diagram (or)
- Right-click on the *Start Event* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>Start Event</i>	
Event Type	Select a trigger for the <i>Start Event</i> , namely: <ul style="list-style-type: none"> • Compensation • Conditional • Error • Escalation • Message • Multiple 	

	<ul style="list-style-type: none"> • None • Parallel Multiple • Signal • Timer 																			
Details	<p>Depending on the selected Event Type, the Details tab changes as follows :</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Message Event Type</td> </tr> <tr> <td>Pool Package</td> <td>Select the package that represents one of the involved <i>Participants</i></td> </tr> <tr> <td>Operation</td> <td>Select an Operation from the list of operations selected Pool Package</td> </tr> <tr> <td colspan="2">Timer Event Type</td> </tr> <tr> <td>Time Cycle</td> <td>Specify the value for the time duration</td> </tr> <tr> <td>Time Date</td> <td>Specify the value for the time date</td> </tr> <tr> <td colspan="2">Error Event Type</td> </tr> <tr> <td>Error</td> <td>Select an <i>Error</i> element</td> </tr> </tbody> </table>	Field	Usage	Message Event Type		Pool Package	Select the package that represents one of the involved <i>Participants</i>	Operation	Select an Operation from the list of operations selected Pool Package	Timer Event Type		Time Cycle	Specify the value for the time duration	Time Date	Specify the value for the time date	Error Event Type		Error	Select an <i>Error</i> element	
Field	Usage																			
Message Event Type																				
Pool Package	Select the package that represents one of the involved <i>Participants</i>																			
Operation	Select an Operation from the list of operations selected Pool Package																			
Timer Event Type																				
Time Cycle	Specify the value for the time duration																			
Time Date	Specify the value for the time date																			
Error Event Type																				
Error	Select an <i>Error</i> element																			
Assignments	Select one or more <i>Assignment</i> elements created in the SupportingElements package in this tab (Optional)	Assignment ^[1267] SupportingElements ^[1256]																		
OK	Save the values entered in the dialog																			
Cancel	Discard the values entered in the dialog																			
Help	Display this Help topic																			
General	Open the UML Properties dialog	Properties ^[660]																		

Notes:

- Either set **Time Cycle** or **Time Date** for **Timer Event Type** but not both as they are mutually exclusive fields.
- **Assignments** tab is not available on *Start Events* that are used to start an *Event Sub-Process*
- **Compensation, Error** and **Timer** event types are valid only on an *Event Sub-Process*
- **Conditional, Escalation, Multiple, Parallel Multiple** and **Signal** event types cannot be mapped to **BPEL 2.0**

8.3.5.2.2 Intermediate Event

A **BPMN 2.0 Intermediate Event** indicates where an event occurs somewhere between the start and end of a *Process*.

BPEL Properties:

- Double-click on the *Intermediate Event* in the **BPEL** diagram

- Right-click on the *Intermediate Event* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also																						
Name	Specify the name for the <i>Intermediate Event</i>																							
Event Type	Select the type for the <i>Intermediate Event</i> , namely: <ul style="list-style-type: none"> Cancel Compensation Conditional Error Escalation Link Message Multiple None Parallel Multiple Signal Timer 																							
Details	Depending on the selected Event Type , the Details tab changes as follows : <table border="1" data-bbox="496 1059 1214 2022"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Message Event Type</td> </tr> <tr> <td>Pool Package</td> <td>Select the package that represents one of involved <i>Participants</i></td> </tr> <tr> <td>Operation</td> <td>Select an Operation from the list of operations under the selected Pool Package</td> </tr> <tr> <td colspan="2">Link Event Type</td> </tr> <tr> <td>Link Event</td> <td>Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i>) from the list of <i>Link Event</i> elements under the current <i>Process / Sub-Process</i></td> </tr> <tr> <td colspan="2">Timer Event Type</td> </tr> <tr> <td>Time Cycle</td> <td>Specify the value for the time duration</td> </tr> <tr> <td>Time Date</td> <td>Specify the value for the time date</td> </tr> <tr> <td colspan="2">Compensation Event Type</td> </tr> <tr> <td>Activity</td> <td>Select an <i>Activity</i> from the list of <i>Activity</i> elements under the current <i>Process/Sub-Process</i></td> </tr> </tbody> </table>	Field	Usage	Message Event Type		Pool Package	Select the package that represents one of involved <i>Participants</i>	Operation	Select an Operation from the list of operations under the selected Pool Package	Link Event Type		Link Event	Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i>) from the list of <i>Link Event</i> elements under the current <i>Process / Sub-Process</i>	Timer Event Type		Time Cycle	Specify the value for the time duration	Time Date	Specify the value for the time date	Compensation Event Type		Activity	Select an <i>Activity</i> from the list of <i>Activity</i> elements under the current <i>Process/Sub-Process</i>	
Field	Usage																							
Message Event Type																								
Pool Package	Select the package that represents one of involved <i>Participants</i>																							
Operation	Select an Operation from the list of operations under the selected Pool Package																							
Link Event Type																								
Link Event	Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i>) from the list of <i>Link Event</i> elements under the current <i>Process / Sub-Process</i>																							
Timer Event Type																								
Time Cycle	Specify the value for the time duration																							
Time Date	Specify the value for the time date																							
Compensation Event Type																								
Activity	Select an <i>Activity</i> from the list of <i>Activity</i> elements under the current <i>Process/Sub-Process</i>																							

	<table border="1"> <tr> <td colspan="2">Error Event Type</td> </tr> <tr> <td>Error</td> <td>Select an <i>Error</i> element</td> </tr> </table>	Error Event Type		Error	Select an <i>Error</i> element	
Error Event Type						
Error	Select an <i>Error</i> element					
Assignments	Select one or more <i>Assignment</i> elements created in the SupportingElements package in this tab (Optional)	Assignment ^[1267] SupportingElements ^[1258]				
OK	Save the values entered in the dialog					
Cancel	Discard the values entered in the dialog					
Help	Display this Help topic					
General	Open the UML Properties dialog	Properties ^[660]				

Notes:

- **Cancel, Conditional, Escalation, Multiple, None, Parallel Multiple** and **Signal** event type cannot be mapped to **BPEL 2.0**
- **Assignments** tab is not available on *Intermediate Events* that are attached to the boundary of an *Activity*
- Either set **Time Cycle** or **Time Date** for **Timer Trigger Type** but not both as they are mutually exclusive fields.
- **Error** event type is valid only on an *Intermediate Event* attached to the boundary of an **Activity**
- **Compensation** event type is valid only on an *Event Sub-Process* or when attached to the boundary of an *Activity*
- **Link Intermediate Event** can be used either as a *GOTO* or an *off-page* connector. So, this event can have either incoming or outgoing *Sequence Flows* - but not both.

8.3.5.2.2.3 Activity

A **BPMN 2.0 Activity** represents work that is performed within a *Process*. An *Activity* can be modeled as a:

- **Sub-Process** - a compound *Activity* that is defined as a flow of other **BPMN 2.0** elements or
- **Task** - an atomic *Activity* that cannot be broken down into a smaller unit.

Activities - both *Tasks* and *Sub-Processes* - can also, optionally, act as *Looping constructs*. The **OMG BPMN 2.0 Specification** defines two types of *Looping* construct:

- **Standard Loop** (*while* or *until*)
- **Multi-Instance Loop** (*for each*)

BPEL Properties:

- Double-click on the *Activity* in the **BPEL** diagram
- Right-click on the *Activity* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>Activity</i>	

Type	Specify whether the <i>Activity</i> is a : <ul style="list-style-type: none"> • Task (or) • Sub-Process 									
Task Type / SubProcess Type	Depending on the value selected in the Type field, the Task Type / SubProcess Type field has the following values : <p>Task Type :</p> <ul style="list-style-type: none"> • Abstract • Business Rule • Manual • Receive • Script • Send • Service • User <p>SubProcess Type :</p> <ul style="list-style-type: none"> • Ad-Hoc • Call Activity • Compensation • Embedded • Event • Transaction 									
Details	Depending on the selected Task Type , the Details tab changes as follows : <table border="1" data-bbox="496 1137 1209 1420"> <thead> <tr> <th data-bbox="496 1137 740 1196">Field</th> <th data-bbox="740 1137 1209 1196">Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2" data-bbox="496 1196 1209 1249">Receive / Send / Service Task Type</td> </tr> <tr> <td data-bbox="496 1249 740 1339">Pool Package</td> <td data-bbox="740 1249 1209 1339">Select the package that represents one of involved <i>Participants</i></td> </tr> <tr> <td data-bbox="496 1339 740 1420">Operation</td> <td data-bbox="740 1339 1209 1420">Select an Operation from the list of operations of the selected Pool Package</td> </tr> </tbody> </table>	Field	Usage	Receive / Send / Service Task Type		Pool Package	Select the package that represents one of involved <i>Participants</i>	Operation	Select an Operation from the list of operations of the selected Pool Package	
Field	Usage									
Receive / Send / Service Task Type										
Pool Package	Select the package that represents one of involved <i>Participants</i>									
Operation	Select an Operation from the list of operations of the selected Pool Package									
Assignments	Select one or more <i>Assignment</i> elements created in the SupportingElements package in this tab (Optional)	Assignment ^[1267] SupportingElements ^[1256]								
Loop Details	Activities can be repeated sequentially, behaving like a loop. Specify the <i>Activity</i> looping details in this tab (Optional) <table border="1" data-bbox="496 1653 1209 2033"> <thead> <tr> <th data-bbox="496 1653 740 1711">Field</th> <th data-bbox="740 1653 1209 1711">Usage</th> </tr> </thead> <tbody> <tr> <td data-bbox="496 1711 740 1839">Loop Type</td> <td data-bbox="740 1711 1209 1839">Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop • Multinstance for creating a <i>for each</i> loop </td> </tr> <tr> <td data-bbox="496 1839 740 1951">Evaluate condition at the beginning of loop</td> <td data-bbox="740 1839 1209 1951">Check this option to create a <i>while Standard</i> loop Default: <i>until</i> loop</td> </tr> <tr> <td data-bbox="496 1951 740 2033">Loop Condition</td> <td data-bbox="740 1951 1209 2033">Enter a boolean condition for Standard loop or a numeric condition for Multinstance loop</td> </tr> </tbody> </table>	Field	Usage	Loop Type	Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop • Multinstance for creating a <i>for each</i> loop 	Evaluate condition at the beginning of loop	Check this option to create a <i>while Standard</i> loop Default: <i>until</i> loop	Loop Condition	Enter a boolean condition for Standard loop or a numeric condition for Multinstance loop	
Field	Usage									
Loop Type	Select: <ul style="list-style-type: none"> • Standard for creating a <i>while</i> or <i>until</i> loop • Multinstance for creating a <i>for each</i> loop 									
Evaluate condition at the beginning of loop	Check this option to create a <i>while Standard</i> loop Default: <i>until</i> loop									
Loop Condition	Enter a boolean condition for Standard loop or a numeric condition for Multinstance loop									

	Loop Maximum	Specify the maximum value for the loop even for a Standard loop (Optional)	
OK	Save the values entered in the dialog		
Cancel	Discard the values entered in the dialog		
Help	Display this Help topic		
General	Open the UML Properties dialog		Properties ^[660]

Notes:

- **Assignments** tab is not applicable for *Sub-Process*
- **Loop Details** tab is not applicable for *Event Sub-Process*
- **Business Rule, Manual, Script** and **User Task** types cannot be mapped to **BPEL 2.0**
- **Ad-Hoc, Call Activity** and **Transaction Sub-Process** types cannot be mapped to **BPEL 2.0**

8.3.5.2.2.4 Gateway

A **BPMN 2.0 Gateway** control the way in which *Sequence Flows* converge and diverge within a *Process*. They provide a gating mechanism that either allows or blocks a *Sequence Flow*.

BPEL Properties:

- Double-click on the *Gateway* in the **BPEL** diagram
- Right-click on the *Gateway* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also						
Name	Specify the name for the <i>Gateway</i>							
Gateway	Select the type for the <i>Gateway</i> , namely: <ul style="list-style-type: none"> • Complex • Event • Exclusive • Inclusive • Parallel 							
Details	Depending on the selected Gateway , the Details tab changes as follows : <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Event Gateway</td> </tr> <tr> <td>Event Type</td> <td>Set to Exclusive to represent an Exclusive Event Based Gateway</td> </tr> </tbody> </table>	Field	Usage	Event Gateway		Event Type	Set to Exclusive to represent an Exclusive Event Based Gateway	
Field	Usage							
Event Gateway								
Event Type	Set to Exclusive to represent an Exclusive Event Based Gateway							
OK	Save the values entered in the dialog							
Cancel	Discard the values entered in the dialog							
Help	Display this Help topic							

General	Open the UML Properties dialog	Properties 6601
---------	---------------------------------------	------------------------------------

Notes:

- The target of the outgoing *Sequence Flows* of this **Event Exclusive Gateway** must be either an:
 - Receive Task**
 - Message** or **Timer Intermediate Event**
- If a **Message Task** is one of the targets of the outgoing *Sequence Flow* of an **Event Exclusive Gateway**, then a **Message Intermediate Event** cannot be the target of the other outgoing *Sequence Flows* of this **Event Exclusive Gateway**

8.3.5.2.2.5 End Event

A **BPMN 2.0 End Event** indicates where a particular *Process* ends. Every *Process* in Enterprise Architect must end with an End Event.

BPEL Properties:

- Double-click on the *End Event* in the **BPEL** diagram
- Right-click on the *End Event* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also												
Name	Specify the name for the <i>End Event</i>													
Event Type	Select the type for the <i>End Event</i> , namely: <ul style="list-style-type: none"> Cancel Compensation Error Escalation Message Multiple None Signal Terminate 													
Details	Depending on the selected Event Type , the Details tab changes as follows : <table border="1" data-bbox="504 1637 1249 2029"> <thead> <tr> <th data-bbox="504 1637 751 1693">Field</th> <th data-bbox="751 1637 1249 1693">Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2" data-bbox="504 1693 1249 1749">Message Event Type</td> </tr> <tr> <td data-bbox="504 1749 751 1827">Pool Package</td> <td data-bbox="751 1749 1249 1827">Select the package that represents one of the involved <i>Participants</i></td> </tr> <tr> <td data-bbox="504 1827 751 1906">Operation</td> <td data-bbox="751 1827 1249 1906">Select an Operation from the list of operation the selected Pool Package</td> </tr> <tr> <td colspan="2" data-bbox="504 1906 1249 1962"></td> </tr> <tr> <td colspan="2" data-bbox="504 1962 1249 2029">Compensation Event Type</td> </tr> </tbody> </table>	Field	Usage	Message Event Type		Pool Package	Select the package that represents one of the involved <i>Participants</i>	Operation	Select an Operation from the list of operation the selected Pool Package			Compensation Event Type		
Field	Usage													
Message Event Type														
Pool Package	Select the package that represents one of the involved <i>Participants</i>													
Operation	Select an Operation from the list of operation the selected Pool Package													
Compensation Event Type														

	<table border="1"> <tr> <td>Activity</td> <td>Select an <i>Activity</i> from the list of <i>Activity</i> elem under the current <i>Process/Sub-Process</i></td> </tr> <tr> <td colspan="2">Error Event Type</td> </tr> <tr> <td>Error</td> <td>Select an <i>Error</i> element</td> </tr> </table>	Activity	Select an <i>Activity</i> from the list of <i>Activity</i> elem under the current <i>Process/Sub-Process</i>	Error Event Type		Error	Select an <i>Error</i> element	
Activity	Select an <i>Activity</i> from the list of <i>Activity</i> elem under the current <i>Process/Sub-Process</i>							
Error Event Type								
Error	Select an <i>Error</i> element							
OK	Save the values entered in the dialog							
Cancel	Discard the values entered in the dialog							
Help	Display this Help topic							
General	Open the UML Properties dialog	Properties [660]						

Notes:

- **Compensation** event type is valid only on an *Event Sub-Process*
- **Cancel, Escalation, Multiple** and **Signal** event types cannot be mapped to **BPEL 2.0**

8.3.5.2.2.6 Data Object

A **BPMN 2.0 Data Object** is mapped to to **BPEL 2.0 Variable** and occurs in the context of a *Process* or *Sub-Process*, just like a [Property](#) [1268]. A *Data Object* cannot have any incoming or outgoing *Sequence Flow* or *Message Flow*.

BPEL Properties:

- Double-click on the *Data Object* in the **BPEL** diagram
- Right-click on the *Data Object* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>Data Object</i>	
Type	Select an XML Schema basic type from the drop-down list (or) use the (...) button to select XML Schema ComplexType, SimpleType, Union or Enumeration	
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties [660]

8.3.5.2.2.7 Property

A **BPMN 2.0 Property** is mapped to **BPEL 2.0 Variable** and occurs in the context of a *Process* or *Sub-Process*, just like a [Data Object](#)^[1265]. But, unlike a *Data Object*, *Property* is not displayed on a **BPEL** diagram. EA supports *Property* on **BPMN 2.0 BPEL** and **BPMN 2.0 Activity** elements.

Access: View | Toolbox > More tools | BPMN 2.0 | BPMN 2.0 Business Process | BPMN 2.0 Types

BPEL Properties: Right-click on the *Property* in the **Project Browser** and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>Property</i>	
Type	Select an XML Schema basic type from the drop-down list (or) use the (...) button to select XML Schema ComplexType , <i>SimpleType</i> , <i>Union</i> or <i>Enumeration</i>	
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties [696]

8.3.5.2.2.8 Sequence Flow

A **BPMN 2.0 Sequence Flow** connector shows the order in which the activities (*Tasks* and *Events*) are performed in a *BPEL Process*.

BPEL Properties:

- Double-click on the *Sequence Flow* in the **BPEL** diagram
- Right-click on the *Sequence Flow* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also
Condition Type	Specify the type of the condition on the <i>Sequence Flow</i> , namely : <ul style="list-style-type: none"> • None • Default • Expression 	
Expression	This field is enabled when the Condition Type is set to Expression Specify a boolean expression to act as a gating condition.	
Ordering	This field is enabled when the Condition Type is set to Expression Specify a numerical value that determines the order in which the condition set in the Expression field is to be evaluated.	
OK	Save the values entered in the dialog	

Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties [660]

8.3.5.2.2.9 Pool

A **BPMN 2.0 Pool** represents a **Participant** in a *Process* and does not map to any specific **BPEL 2.0** element. Enterprise Architect uses *Pools* to represent external **Participants**, with which the *BPEL Process* communicates. These are 'black box' *pools* i.e., they are abstract and do not expose any details (they do not contain any **BPMN 2.0** elements inside them).

BPEL Properties:

- Double-click on the *Pool* in the **BPEL** diagram
- Right-click on the *Pool* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>Pool</i>	
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties [660]

Notes:

- A *BPEL Process* should not contain a *Pool* as its child element.
- A **BPEL** diagram under a *BPEL Process* contains an implicit *Pool* - so it is invalid to add a *Pool* in this **BPEL** diagram to represent the *BPEL Process*.
- *Pool* cannot have any incoming or outgoing *Sequence Flow* connectors - it can have only incoming or outgoing *Message Flow* connectors.

8.3.5.2.2.10 Assignment

A **BPMN 2.0 Assignment** element enables data to be copied between messages and variables within a *BPEL Process*. An *Assignment* element maps to a **BPEL 2.0 assign** activity and copies the specified value from the source to the target.

In Enterprise Architect, *Assignment* elements should be created in the **Assignments** package in **SupportingElements** [1258]. If they are created elsewhere, they cannot be enacted correctly.

Access: [View](#) | [Toolbox > More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Business Process](#) | [BPMN 2.0 Types](#)

BPEL Properties:

- Double-click on the *Assignment* in the **BPEL** diagram
- Right-click on the *Assignment* in the **BPEL** diagram (or the **Project Browser**) and select the following context menu : **BPEL | BPEL 2.0 Properties**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the <i>Assignment</i>	
Copy From		
Type	Depending on the value selected in this field, further details are required	
Literal	Specify a literal value	
Expression	Specify an expression	
Message	Select a BPMN 2.0 Message in the package representing the <i>BPEL Process / Pool</i> under the SupportingElements package	SupportingElement s ^[1256] Web Service Operation ^[1268]
Part	Select a BPMN 2.0 Property belonging to the selected Message	
Variable	Select a BPMN 2.0 Data Object or BPMN 2.0 Property created under the <i>BPEL Process</i>	
Copy To		
Message	Select a BPMN 2.0 Message in the package representing the <i>BPEL Process / Pool</i> under the SupportingElements package	SupportingElement s ^[1256] Web Service Operation ^[1268]
Part	Select a BPMN 2.0 Property belonging to the selected Message	
Variable	Select a BPMN 2.0 Data Object or BPMN 2.0 Property created under the <i>BPEL Process</i>	
OK	Save the values entered in the dialog	
Cancel	Discard the values entered in the dialog	
Help	Display this Help topic	
General	Open the UML Properties dialog	Properties ^[660]

Notes:

- Messages are created when you create a **Web Service Operation**

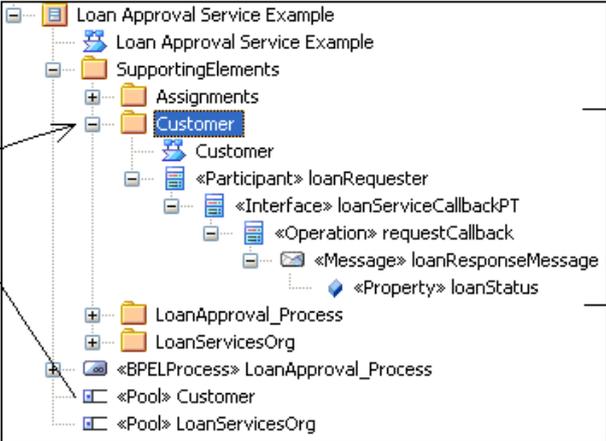
8.3.5.2.3 Create BPEL 2.0 Web Service Operation

BPEL is an **Orchestration** language, which orchestrates services that are exposed using **WSDL 1.1**. It coordinates the execution of the various operations of these involved services. **BPEL 2.0** supports **WSDL 1.1 one-way** and **request-response** operations only (and not **solicit-response** and **notification** operations). Enterprise Architect enables you to create, for *BPEL Process* and *Pool* elements, **Web Service Operations** that support either *synchronous* (request-response) or *asynchronous* (one-way) interactions.

Access: Right-click on a BPEL Process or Pool and select the following context menu: **BPEL | Create WebService**

Reference:

Field/Button	Usage	See Also																
Operation	Specify whether to create a new Web Service Operation or create one from an existing WSDL created / imported into EA (using the Import WSDL dialog) by selecting either : <ul style="list-style-type: none"> • Create New (or) • Create from existing WSDL PortType Operation 	Model WSDL ^[1612] Import WSDL ^[1622]																
Type	Specify whether to create a <i>one-way</i> or <i>request-response</i> Operation by selecting either : <ul style="list-style-type: none"> • Asynchronous (or) • Synchronous 																	
Partnership Details	<p>The interaction between the Web Service and the <i>BPEL Process</i> is modeled as a BPEL partnerLink. For this interaction, provide a:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Name of the BPEL partnerLink</td> </tr> <tr> <td>Role</td> <td>Name of either <i>myRole</i> (if this operation belongs to the <i>BPEL Process</i> Web Service) or <i>partnerRole</i> (if this operation belongs to the Web Service Pool) of the BPEL partnerLink.</td> </tr> </tbody> </table>	Field	Usage	Name	Name of the BPEL partnerLink	Role	Name of either <i>myRole</i> (if this operation belongs to the <i>BPEL Process</i> Web Service) or <i>partnerRole</i> (if this operation belongs to the Web Service Pool) of the BPEL partnerLink .											
Field	Usage																	
Name	Name of the BPEL partnerLink																	
Role	Name of either <i>myRole</i> (if this operation belongs to the <i>BPEL Process</i> Web Service) or <i>partnerRole</i> (if this operation belongs to the Web Service Pool) of the BPEL partnerLink .																	
Web Service	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">When Create New is selected in Operation field:</td> </tr> <tr> <td>PortType</td> <td>Name of the WSDL 1.1 PortType (Interface)</td> </tr> <tr> <td>Operation</td> <td>Name of the WSDL 1.1 PortType Operation</td> </tr> <tr> <td colspan="2">When Create from existing WSDL PortType Operation is selected in Operation field:</td> </tr> <tr> <td>WSDL Package</td> <td>Select an existing WSDL package created / imported into EA</td> </tr> <tr> <td>PortType</td> <td>Select a WSDL 1.1 PortType from the selected WSDL Package</td> </tr> <tr> <td>Operation</td> <td>Select a WSDL 1.1 PortType Operation in the selected PortType</td> </tr> </tbody> </table>	Field	Usage	When Create New is selected in Operation field:		PortType	Name of the WSDL 1.1 PortType (Interface)	Operation	Name of the WSDL 1.1 PortType Operation	When Create from existing WSDL PortType Operation is selected in Operation field:		WSDL Package	Select an existing WSDL package created / imported into EA	PortType	Select a WSDL 1.1 PortType from the selected WSDL Package	Operation	Select a WSDL 1.1 PortType Operation in the selected PortType	
Field	Usage																	
When Create New is selected in Operation field:																		
PortType	Name of the WSDL 1.1 PortType (Interface)																	
Operation	Name of the WSDL 1.1 PortType Operation																	
When Create from existing WSDL PortType Operation is selected in Operation field:																		
WSDL Package	Select an existing WSDL package created / imported into EA																	
PortType	Select a WSDL 1.1 PortType from the selected WSDL Package																	
Operation	Select a WSDL 1.1 PortType Operation in the selected PortType																	
Input	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">When Create New is selected in Operation field:</td> </tr> </tbody> </table>	Field	Usage	When Create New is selected in Operation field:														
Field	Usage																	
When Create New is selected in Operation field:																		

	<table border="1"> <tr> <td>Message Name</td> <td>Name of the WSDL 1.1 Message</td> </tr> <tr> <td>Properties</td> <td>Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type</td> </tr> <tr> <td colspan="2"> <p>When Create from existing WSDL PortType Operation is selected in the Operation field:</p> <p>The fields in this tab are pre-filled with the details of the input WSDL 1.1 Message (of the WSDL 1.1 PortType Operation selected in the Operation field in the Web Service tab)</p> </td> </tr> </table>	Message Name	Name of the WSDL 1.1 Message	Properties	Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type	<p>When Create from existing WSDL PortType Operation is selected in the Operation field:</p> <p>The fields in this tab are pre-filled with the details of the input WSDL 1.1 Message (of the WSDL 1.1 PortType Operation selected in the Operation field in the Web Service tab)</p>						
Message Name	Name of the WSDL 1.1 Message											
Properties	Click on the (...) button to enter the WSDL 1.1 Message Part Name and XSD Type											
<p>When Create from existing WSDL PortType Operation is selected in the Operation field:</p> <p>The fields in this tab are pre-filled with the details of the input WSDL 1.1 Message (of the WSDL 1.1 PortType Operation selected in the Operation field in the Web Service tab)</p>												
<p>Output</p>	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2"> <p>When Create New is selected in Operation field:</p> </td> </tr> <tr> <td>Message Name</td> <td>Name of the WSDL 1.1 Message</td> </tr> <tr> <td>Properties</td> <td>Press (...) button to enter the WSDL 1.1 Message Part Name and XSD Type</td> </tr> <tr> <td colspan="2"> <p>When Create from existing WSDL PortType Operation is selected in the Operation field:</p> <p>The fields in this tab are pre-filled with the details of the output WSDL 1.1 Message (of the WSDL 1.1 PortType Operation selected in the Operation field in the Web Service tab)</p> </td> </tr> </tbody> </table>	Field	Usage	<p>When Create New is selected in Operation field:</p>		Message Name	Name of the WSDL 1.1 Message	Properties	Press (...) button to enter the WSDL 1.1 Message Part Name and XSD Type	<p>When Create from existing WSDL PortType Operation is selected in the Operation field:</p> <p>The fields in this tab are pre-filled with the details of the output WSDL 1.1 Message (of the WSDL 1.1 PortType Operation selected in the Operation field in the Web Service tab)</p>		
Field	Usage											
<p>When Create New is selected in Operation field:</p>												
Message Name	Name of the WSDL 1.1 Message											
Properties	Press (...) button to enter the WSDL 1.1 Message Part Name and XSD Type											
<p>When Create from existing WSDL PortType Operation is selected in the Operation field:</p> <p>The fields in this tab are pre-filled with the details of the output WSDL 1.1 Message (of the WSDL 1.1 PortType Operation selected in the Operation field in the Web Service tab)</p>												
<p>OK</p>	<p>Create a Web Service Operation based on the values entered in the dialog.</p> <p>The operation is created in a package that has the same name as the <i>BPEL Process / Pool</i> from from which this dialog is invoked, under the SupportingElements package</p> 	<p>SupportingElements <small>1256</small></p>										
<p>Cancel</p>	<p>Discard the values entered in the dialog and abort creating the Web Service Operation</p>											

Help	Display this Help topic	
------	-------------------------	--

Notes:

- The Output tab is not applicable for an Asynchronous operation

8.3.5.2.4 Generate BPEL 2.0

BPEL 2.0 code can be generated from a *BPEL Process*. Enterprise Architect validates the *BPEL Process* before generating the **BPEL 2.0** code. In addition to generating the **BPEL 2.0** code, **WSDL 1.1** files are generated for the *BPEL Process* and all the involved *Pools* (provided **Web Service Operations** are defined for them).

Access: Right-click on a BPEL Process and select the following context menu: **BPEL | Generate BPEL 2.0**

Reference:

Field/Button	Usage	See Also
File Name	Specify the path where the BPEL 2.0 file is to be generated	
Namespace Details	Double-click on an entry (if any) in this field to open the Namespace Details dialog and add / edit the namespace details The entry DefaultPool represents the current <i>BPEL Process</i>	
Generate BPEL	Validate the model and generate BPEL 2.0	BPEL Model Validation ^[1271]
Close	Close this dialog	
Help	Display this Help topic	
View BPEL	View the generated BPEL 2.0 file	

Learn More:

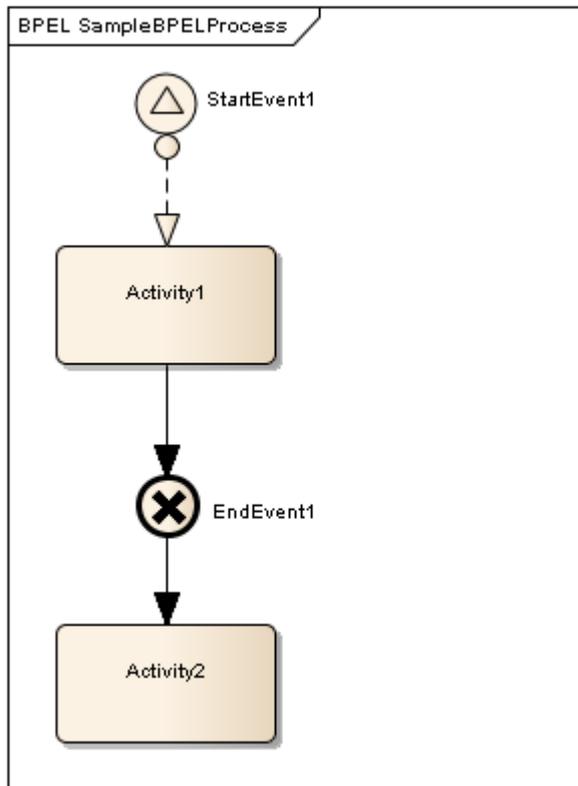
- [Model a BPEL 2.0 Process](#) ^[1257]
- [Create a Web Service Operation](#) ^[1265]

8.3.5.3 BPEL Model Validation

The **BPEL** model is validated for both *syntax* and *semantic* errors before generating the **BPEL** code. Upon successful validation, the **BPEL** file (and the associated **WSDL 1.1** files, if any) are generated. The results of the validation, as well as the progress of the **BPEL** generation, are displayed in the BPEL Progress tab of the Output window.

Example BPEL Model Violation

The following model shows several basic **BPEL** violations:



1. *StartEvent1* is of type **Signal**, which cannot be mapped to **BPEL**.
2. The *Message Flow* connector between *StartEvent1* and *Activity1* is invalid, as *Message Flow* cannot be used to connect *Activities* or *Events* within a *Process*.
3. *EndEvent1* is of type **Cancel**, which cannot be mapped to **BPEL**.
4. *EndEvent1* cannot have any outgoing *Sequence Flow* connector, as it represents the end of a *Process*.
5. *Activity2* is not valid, as only an *End Event* represents the end of a *Process*.

Notes:

- Double-click on a validation error entry in the BPEL Progress tab of the Output window to go to the source of the error in the Project Browser

Learn More:

- [Generate BPEL 1.1](#)^[1253]
- [BPEL 1.1 Modeling Restrictions](#)^[1236]
- [Generate BPEL 2.0](#)^[1271]
- [BPEL 2.0 Modeling Restrictions](#)^[1254]
- [System Output Window](#)^[127]

8.3.6 ArchiMate

ArchiMate is an open-standard enterprise architecture language from The Open Group, based on the IEEE 1471 standard. It offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure, enabling Enterprise Architects to describe, analyse and visualize the relationships among business domains in an unambiguous way.

Access: From the Toolbox select: **More Tools | ArchiMate**

Topics:

Images	Topic	Detail	See also
	ArchiMate in Enterprise Architect	Enterprise Architect enables you to develop ArchiMate diagrams quickly and simply, through use of an ArchiMate MDG Technology integrated with the Enterprise Architect installer. The ArchiMate facilities are provided in the form of: <ul style="list-style-type: none"> • An ArchiMate diagram type, accessed through the New Diagram dialog • A set of ArchiMate pages in the Toolbox • ArchiMate element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	New Diagram ^[568] Toolbox Shortcut ^[55] Quick Linker ^[622]
	ArchiMate Toolbox Pages	The toolbox pages provide three categories of elements - Information, Behavior and Structure - and a page of connectors that are largely based on the UML connectors. <p>The appearance of elements can be modified with the use of Tagged Values, as suggested below:</p> <ul style="list-style-type: none"> • For <i>Artifact, Process, Function, Interaction, Service, Event, Actor, Role, Collaboration, Interface, Component, Node</i> and <i>Device</i> elements: • <i>iconstyle=true</i> shows the iconic representation of the element • <i>iconstyle=false</i> shows a rectangle with a decoration in the top corner • An Interface element (if <i>iconstyle=true</i>) can be provided, required, symmetric or assembly, and can be rotated, by setting the Tagged Values • A Function element (if <i>iconstyle=true</i>) can be rotated • An Actor element (if <i>iconstyle=false</i>) can be compound • Process and Function elements can be marked <i>atomic</i> Because there is a large range of ArchiMate elements, you should make good use of the Quick Linker arrow to guide you in selecting appropriate source and target elements and relationship types to model your enterprise architecture.	
	Disable ArchiMate	If you prefer not to use ArchiMate in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	MDG Technologies ^[1033]

8.3.7 Eriksson-Penker Extensions

Eriksson-Penker extensions (developed by H. E. Eriksson and M.Penker) provide a framework for UML business processing model extensions, to which an enterprise architect can add stereotypes and properties appropriate to their business.

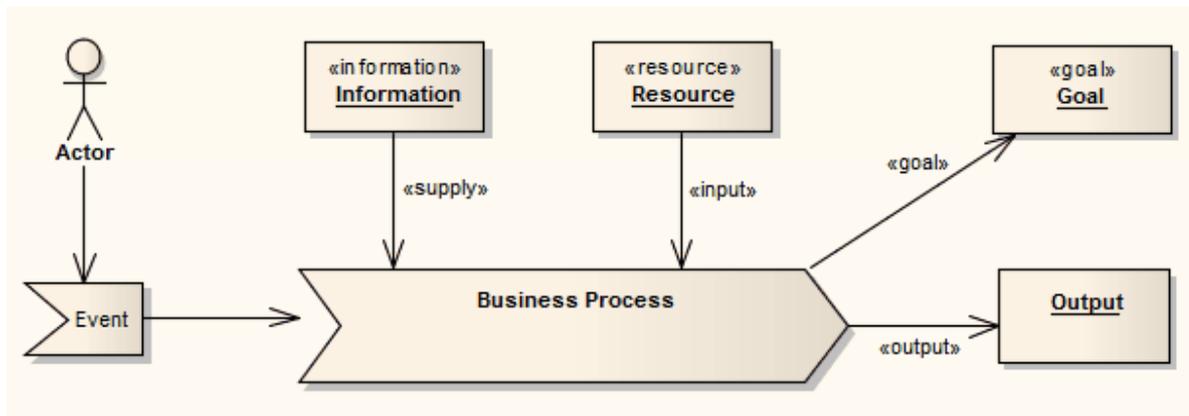
Access: From the Toolbox select: **More Tools | Eriksson-Penker Extensions**

Topics:

Images	Topic	Detail	See also
	Eriksson-Penker in Enterprise Architect	<p>Enterprise Architect provides - through the integration of MDG Technologies with the installer - two well-respected and proven UML extensions that further enhance the capture of business activities, processes, objects and information flows. One of these is Business Process Modeling Notation (BPMN). The other is the Eriksson-Penker profile which, through a set of stereotypes, provides a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization.</p> <p>The Eriksson-Penker extensions are provided in the form of:</p> <ul style="list-style-type: none"> • An Eriksson-Penker diagram type, accessed through the New Diagram dialog • An Eriksson-Penker page in the Toolbox • Eriksson-Penker element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	Business Process Modeling Notation ^[1219] New Diagram ^[568] Toolbox Shortcut ^[551] Quick Linker ^[622]
	Eriksson-Penker Toolbox Page	Please see image to the left .	

Example:

The following is an example of a simple Eriksson-Penker diagram:



Learn More:

- [The Business Process Model](#) (Online Resource)

8.4 Data Models

Data Modelers and Information Architects are challenged with creating data models that span multiple levels of abstraction - from concept to physical implementation. They might also be responsible for maintaining traceability between these models. Enterprise Architect helps to meet these challenges with easy-to-use tools for building and maintaining all of the fundamental data models: Conceptual, Logical and Physical data models.

Topics:

Topic	Detail	See also
Conceptual Data Models	<p>Conceptual data models, also called Domain models, establish the basic concepts and semantics of a given domain and help to communicate these to a wide audience of stakeholders</p> <p>Conceptual models also serve as a common vocabulary during the analysis stages of a project; they can be created in Enterprise Architect using Entity-Relationship or UML Class models</p>	
Logical Data Models	<p>Logical data models add further detail to conceptual model elements and refine the structure of the domain; they can be defined using UML Class models</p> <p>One benefit of a Logical data model is that it provides a foundation on which to base the Physical model and subsequent database implementation</p>	
Physical Data Models	<p>Physical data models in Enterprise Architect help you visualize your database structure and automatically derive the corresponding database schema; you use Enterprise Architect's UML Profile for Data Modeling specifically for this purpose</p> <p>The profile provides useful extensions of the UML standard that map database concepts of tables and relationships onto the UML concepts of Classes and Associations; you can also model database keys, triggers, constraints, referential integrity and other relational database features</p> <p>Because Enterprise Architect lets you visualize each type of data model in the same repository, you can easily manage dependencies between each level of abstraction; this helps you maximize traceability and verify completeness of system implementation</p>	

Learn more:

- [Conceptual data modeling](#)^[1278]
- [Logical data modeling](#)^[1279]
- [Physical data modeling](#)^[1346]

8.4.1 Table



Description:

A Table is a stereotyped Class. It is drawn with a small table icon in the upper right corner. You typically use this element in Data Modeling diagrams.

A Table element has a special Properties dialog, with settings for database type and the ability to set column information and data-related operations such as triggers and indexes. When setting up a Table, make sure you set the default database type for that Table, otherwise you do not have any data types to choose from when creating columns.

Toolbox Icon:

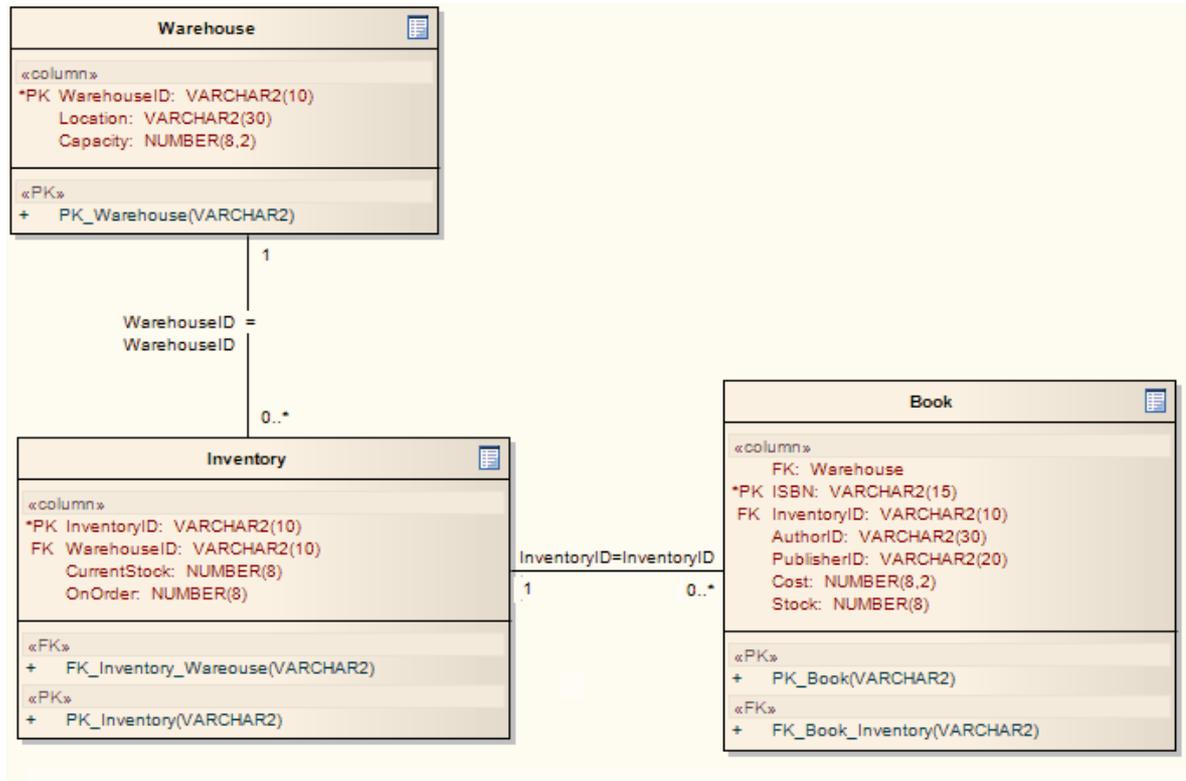


Learn More:

- [Data Modeling Diagram](#)^[1278]
- [Set the Default Database Type](#)^[1348]

8.4.2 Database Schema

The following diagram shows an example Database Schema represented by a **Physical Data Model**.



Learn More:

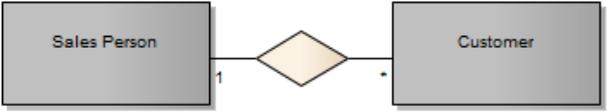
- [Physical Data Model](#)^[1346]
- [Data Modeling Toolbox](#)^[566]

8.4.3 Conceptual Data Model

A Conceptual data model is the most abstract form of data model. It is helpful for communicating ideas to a wide range of stakeholders because of its simplicity. Therefore platform-specific information, such as data types, are omitted from a Conceptual data model. Other implementation details, such as procedures and interface definitions, are also excluded. Below is an example of a Conceptual data model that is rendered using two of the notations supported by Enterprise Architect.

Using Entity-Relationship (ER) notation, we represent the data concepts "Sales Person" and "Customer" as Entities with a 1-many relationship between them.

Topics:

Topic	Detail	See also
<p>Example</p>	<div style="text-align: center;">  </div> <p>A Conceptual data model that uses Entity-Relationship notation</p> <p>We can represent the exactly the same semantic information using UML Classes and Associations</p> <p>Whether you use UML or ER notation to represent data concepts in your project depends on the experience and preferences of the stakeholders involved</p> <div style="text-align: center;">  </div> <p>An equivalent model using UML Class notation</p> <p>The detailed structure of the data concepts illustrated above is defined later by the Logical data model</p>	

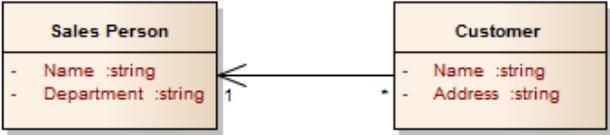
Learn more:

- [Logical data models](#)^[1279]
- [Entity Relationship modeling](#)^[1382]
- [UML Class modeling](#)^[798]

8.4.4 Logical Data Model

Logical data models help to define the detailed structure of the data elements in a system and the relationships between data elements. They refine the data elements introduced by a Conceptual data model and form the basis of the Physical data model. In Enterprise Architect, a Logical data model is typically represented using the UML Class notation.

Topics:

Topic	Detail	See also
<p>Example</p>	<p>A simple example of a Logical data model is provided below</p> <p>Note that the data elements "Sales Person" and "Customer" contain UML Attributes; the Attribute types, however, remain platform-independent</p> <div style="text-align: center;">  </div> <p>A Logical data model defined using UML Class notation</p> <p>Platform-specific Attribute types and other meta-data that relate to</p>	

Topic	Detail	See also
	a specific DBMS implementation are defined by the Physical data model	

Learn more:

- [Physical data models](#) ¹³⁴⁶
- [UML Class modeling](#) ⁷⁹⁸

8.5 User Interface Models

User Interface Diagrams are customized UML diagrams used to visually mock-up a system's user interface using forms, controls and labels.

Example Diagram: [Example User Interface Diagram](#) 

Tools:

Select User Interface diagram elements and connectors from the User Interface pages of the Toolbox.

User Interface Diagram Elements	User Interface Diagram Connectors
 Package	 Associate
 Screen	 Aggregate
 UI Control	 Generalize
 Object	 Realize

Notes:

- The Enterprise Architect Professional, Corporate and Suite editions also include the MDG Win32 UI Technology, which enables you to design user interface components that render more precisely as Win32® User Interface elements

Learn More:

- [Win32 UI Technology](#) 

8.5.1 Screen

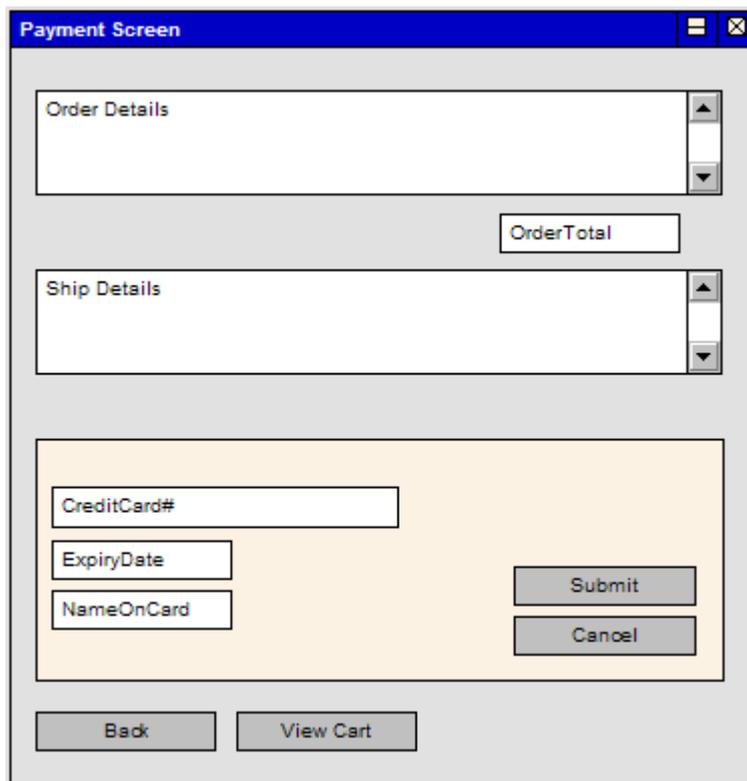
Description

A Screen is used to prototype User Interface screen flow. By using UML features such as requirements, constraints and scenarios against User Interface diagram elements, you can build up a solid and detailed understanding of user interface behavior without having to use code. This becomes an excellent means of establishing the precise behavior the system has from a user perspective, and in conjunction with the Use Case model, defines exactly how a user gets work done.

Web pages can also be prototyped and specified rigorously using Enterprise Architect's custom interface extensions.

Example

The example diagram below illustrates some features of Enterprise Architect's screen modeling extensions that support web page prototyping. By adding requirements, rules, scenarios and notes to each element, a detailed model is built up of the form or web page, without having to resort to GUI builders or HTML.



Enterprise Architect displays [UI Controls](#) ^[1282] as a range of special icons, depending on the stereotype used; for example, a Control stereotyped as a «list» displays with a vertical scroll bar.

Toolbox Icon



Learn More:

- [User Interface Diagram](#) ^[1281]
- [UI Controls](#) ^[1282]

8.5.2 UI Control Elements

Description

A UI Control element represents a user interface control element (such as an edit box). It is used for capturing the components of a screen layout and requirements in a Custom or User Interface diagram.

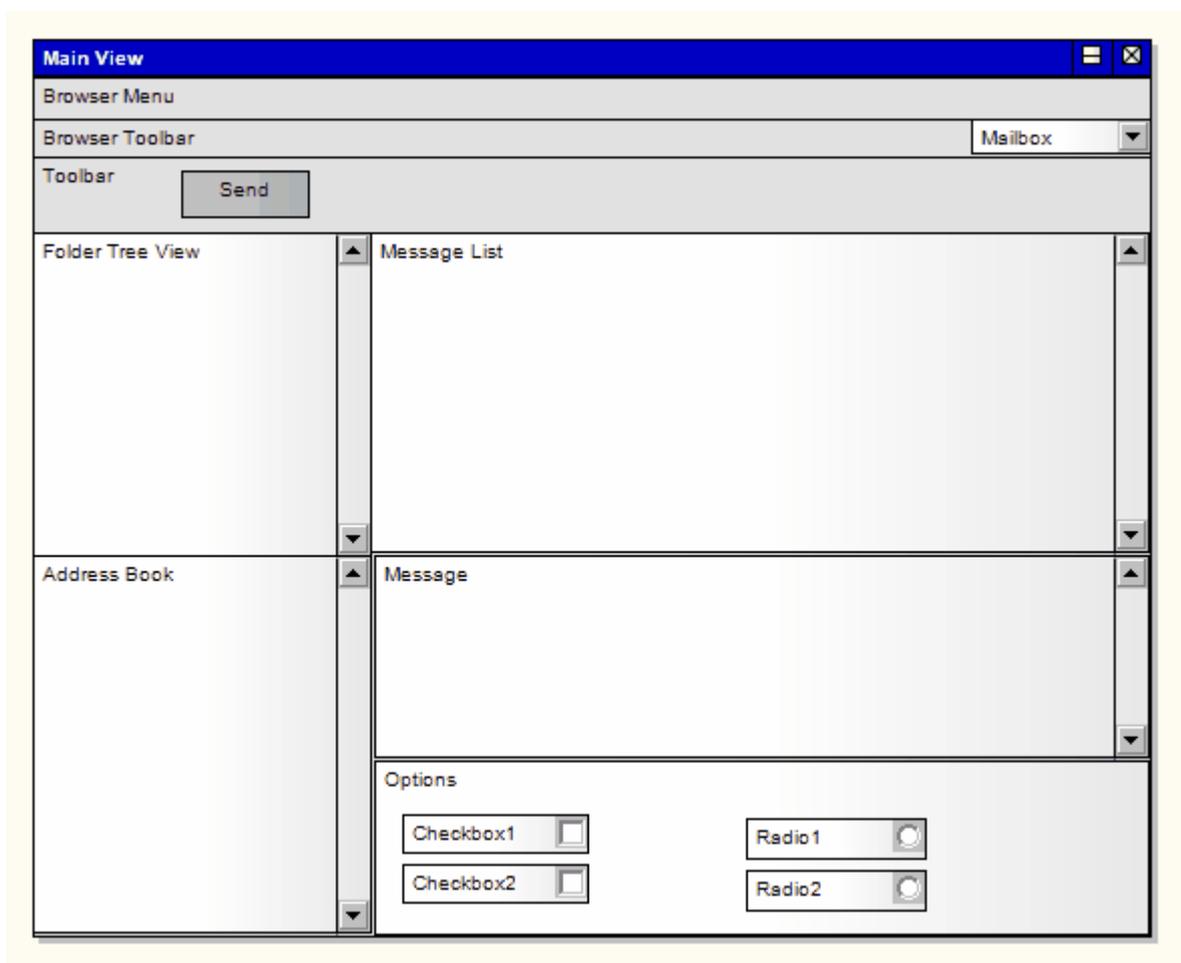
There are a number of UI Control elements available in the User Interface page of the Toolbox. These include:

- List
- Table
- Text Box
- Label
- Form
- Panel

- Button
- Combobox
- Checkbox
- Checkbox (left hand side)
- Radio button
- Radio button (left hand side)
- Vertical Line
- Horizontal Line.

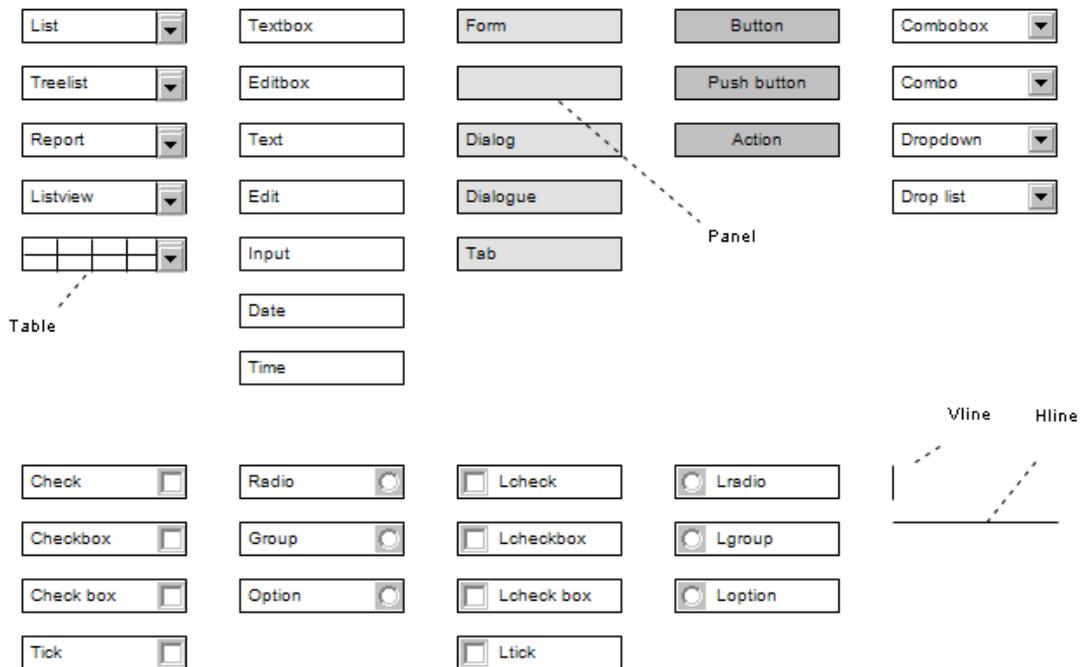
Example

The icons can be combined on a Screen icon to represent the appearance of a user interface screen, as shown:



You can also extend the available icons by selecting other stereotypes in the UI Control Element Properties dialog. The full set of available stereotypes is shown below; type or select the text in the **Stereotype** field to create the corresponding icon.

ui User Interface

Toolbox Icon

(where **UI Control** is the name of the user interface element type)

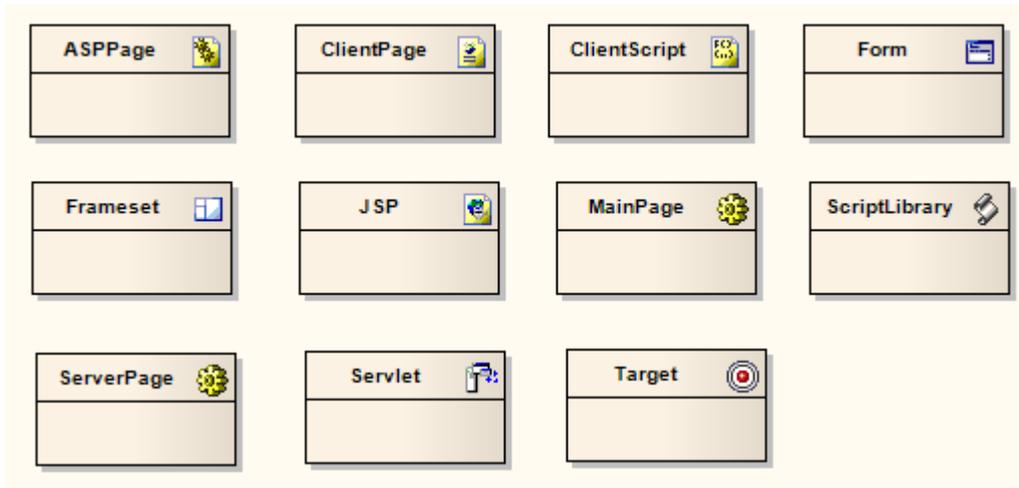
Learn More

- [User Interface](#)^[1281]
- [Screen](#)^[1281]

8.5.3 Web Stereotypes

Description

Enterprise Architect supports a number of stereotypes for web page modeling, the graphical elements for which display with a **graphical icon** instead of the usual «*stereotype*» format. These stereotypes are only supported for Class elements. The image below indicates the various graphical icons and their associated stereotypes.



A similar set of web modeling elements and their relationships are also available through dedicated Web Modeling pages in the Toolbox.

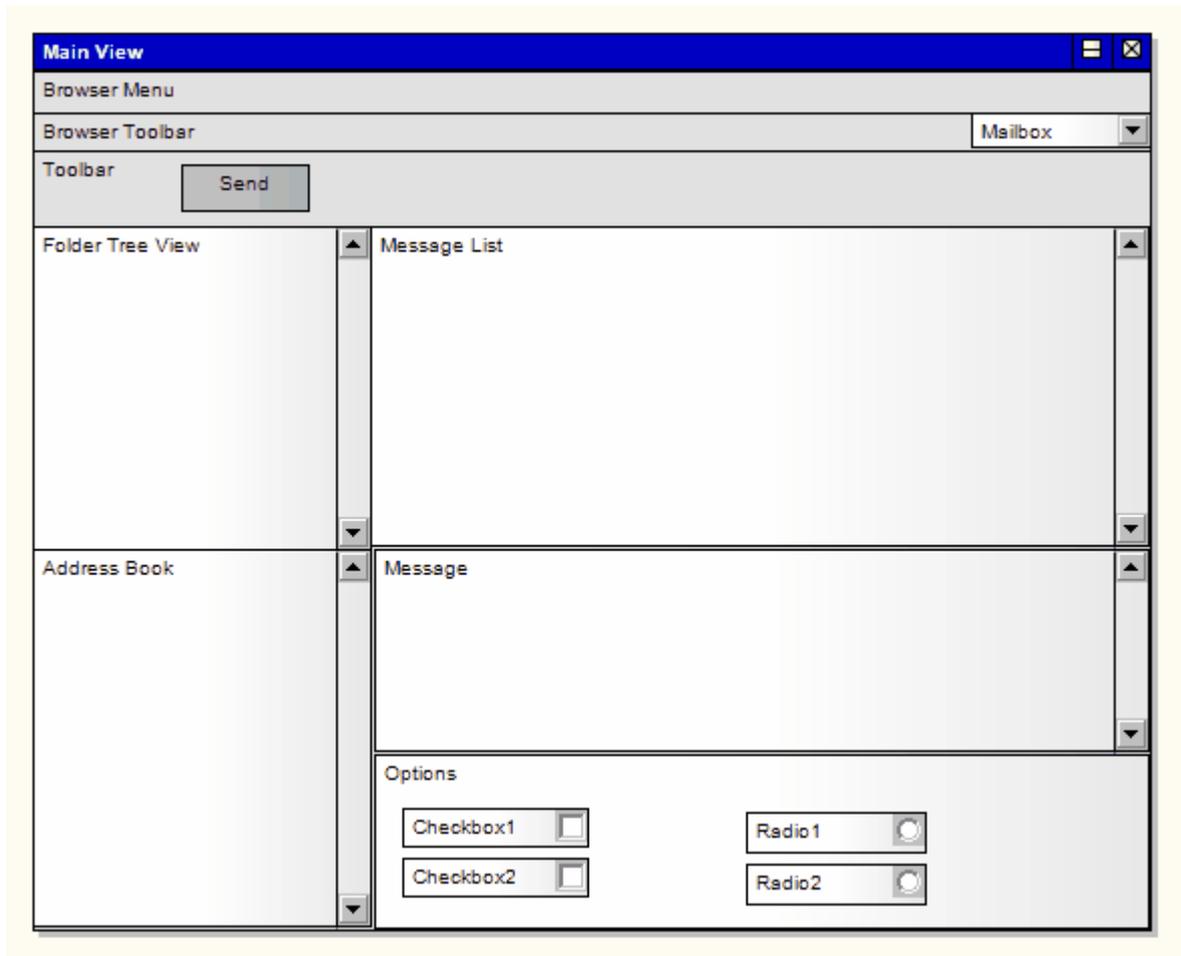
How To:

To set a web icon, follow the steps below:

Step	Action
1	Create a new Class element in a diagram
2	Display the Class Properties dialog
3	In the Stereotype field, either type in the required stereotype name or click on the drop-down arrow and select the required stereotype (as named above)
4	Click on the OK button The Class displays as in one of the examples above

8.5.4 Example User Interface Diagram

In the example User Interface diagram below, forms, controls and labels are arranged on the diagram to describe its appearance. UI Control elements can also be traced to other model elements linking the UI with the underlying implementation.



8.5.5 Win32 UI Technology

The MDG Win32 UI Technology enables you to design user interface screens that render as Win32® user interface elements.

Use To:

- Create and maintain User Interface diagrams in the Win32® User Interface formats

Access: The Win32 UI facilities are provided in the form of:

- A Win32 User Interface diagram type, accessed through the New Diagram dialog
- Win32® User Interface pages in the Toolbox
- Win32 User Interface element and relationship entries in the Toolbox **Shortcut** menu and Quick Linker

Win32 User Interface Toolbox Pages:

You can access the Win32 User Interface Toolbox pages through the **View | Toolbox: More tools... User Interface - Win32** option.

You can also set Win32® User Interface Modeling as the active default technology to access the Toolbox pages directly.

Disable Win32 User Interface Technology:

If you prefer not to use the Win32® UI Technology in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

Notes:

- The MDG Win32® User Interface Technology is available in the Enterprise Architect Professional, Corporate and Suite editions

8.6 Other Stereotypes

There are many other UML elements that you can also work with in Enterprise Architect, most of which are basic elements extended by the use of stereotypes. This topic gives a brief introduction to some of these elements.

- [Analysis Stereotypes](#) ^[1186]
- [Boundary](#) ^[1288]
- [Composite Elements](#) ^[647]
- [Control](#) ^[1289]
- [Entity](#) ^[1290]
- [Event](#) ^[1292]
- [Feature](#) ^[1166]
- [Hyperlinks](#) ^[1292]
- [Image](#) ^[1295]
- [N-Ary Association](#) ^[1296]
- [Packaging Component](#) ^[1297]
- [Process](#) ^[1297]
- [Requirement](#) ^[1154]
- [Risk](#) ^[1298]
- [Screen](#) ^[1281]
- [Task](#) ^[1299]
- [Test Case](#) ^[1672]
- [Test Cut](#) ^[1675]
- [Test Set](#) ^[1675]
- [Test Suite](#) ^[1676]
- [Table](#) ^[1277]
- [UI Control Elements](#) ^[1282]
- [Web Stereotypes](#) ^[1284]

Learn More:

- [UML Stereotypes](#) ^[1016]

8.6.1 Boundary



Description

A Boundary is a stereotyped Object that models some system boundary, typically a user interface screen. You can also create a Boundary as a stereotyped Class. See the [Create a Boundary](#) topic. Boundary elements are used in analysis to capture user interactions, screen flows and element interactions (or 'collaborations').

A Boundary is used in the conceptual phase to capture users interacting with the system at a screen level (or some other boundary interface type). It is often used in Sequence and Robustness (Analysis) diagrams. It is the View in the Model-View-Controller pattern.

Toolbox Icon



Learn More:

- [Create a Boundary](#)^[1289]
- [Analysis Stereotypes](#)^[1186]
- [Sequence Diagram](#)^[849]
- [Communication Diagram](#)^[859]
- [Object Diagram](#)^[799]
- [Analysis Diagram](#)^[1187]

8.6.1.1 Create a Boundary

To create a Boundary element on a diagram as an Object, follow the steps below

Step	Action
1	In the Toolbox , select the More tools Analysis menu option.
2	From the Analysis Elements page, drag the Boundary element onto the diagram.

To create a Boundary element as a stereotyped Class, using the Class Properties dialog, follow the steps below

Step	Action
1	Insert a new Class.
2	Right-click on the element and select the Properties context menu option; the Properties dialog displays.
3	In the Stereotype field, type the value boundary .
4	Click on the Apply and OK buttons.
5	Save the diagram (Ctrl+S).

8.6.2 Control



Description

A Control is a stereotyped Object that models a controlling entity or manager. A Control organizes and schedules other activities and elements, typically in Analysis (including Robustness), Sequence and Communication diagrams. It is the controller of the Model-View-Controller pattern.

You can also create a Control as a stereotyped Class.

Toolbox Icon



Learn More:

- [Create a Control Element](#)^[1290]
- [Analysis Stereotypes](#)^[1186]
 - [Sequence Diagram](#)^[849]
 - [Communication Diagram](#)^[859]
 - [Object Diagram](#)^[799]
 - [Analysis Diagram](#)^[1187]

8.6.2.1 Create a Control Element

How to:

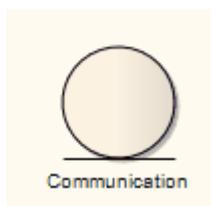
To create a Control element on a diagram as an Object, follow the steps below

Step	Action
1	In the Toolbox, select the More tools Analysis menu option
2	From the Analysis Elements page, drag the Control element onto the diagram

To create a Control element as a stereotyped Class, using the Class Properties dialog, follow the steps below

Step	Action
1	Insert a new Class
2	Right-click on the element and select the Properties context menu option The Properties dialog displays
3	In the Stereotype field, type the value control
4	Click on the Apply and OK buttons
5	Save the diagram (Ctrl+S)

8.6.3 Entity



Description

An Entity is a stereotyped Object that models a store or persistence mechanism that captures the information or knowledge in a system. It is the Model in the Model-View-Controller pattern.

You can also create an Entity as a stereotyped Class. See the Create an Entity topic.

Toolbox Icon



Learn More:

- [Create an Entity](#) ^[1297]
- [Analysis Stereotypes](#) ^[1186]
- [Sequence Diagram](#) ^[849]
- [Communication Diagram](#) ^[859]
- [Object Diagram](#) ^[799]
- [Analysis Diagram](#) ^[1187]

8.6.3.1 Create an Entity

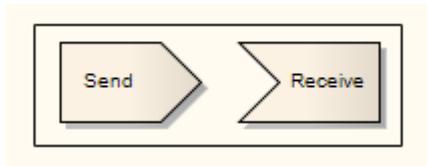
To create an Entity element on a diagram as an Object, follow the steps below

Step	Action
1	In the Toolbox , select the More tools Analysis menu option.
2	From the Analysis Elements page, drag the <i>Entity</i> element onto the diagram.

To create an Entity element as a stereotyped Class, using the Class Properties dialog, follow the steps below

Step	Action
1	Insert a new Class.
2	Right-click on the element and select the Properties context menu option; the Properties dialog displays.
3	In the Stereotype field, type the value entity .
4	Click on the Apply and OK buttons.
5	Save the diagram (Ctrl+S).

8.6.4 Event



Description:

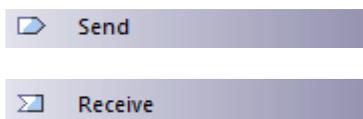
Two elements are used to model *Events*; the:

- *Send Event* which models the generation of a stimulus in the system and the passing of that stimulus to other elements, either within the system or external to the system
- *Receive Event*, depicted as a rectangle with a recessed 'V' on the left side, which indicates that an event occurs in the system due to some external or internal stimulus; typically this invokes further activities and processing

Send and Receive Events can be added from the Analysis and Activity Element pages of the Toolbox.

If you should select the wrong type of event, or otherwise want to change the type, right-click on the Event and select the **Advanced | Make Sender** or **Advanced | Make Receiver** context menu option, as appropriate.

Toolbox Icon:



8.6.5 Hyperlinks

Description

You can place a *Hyperlink* element onto a diagram. This element is a type of text element, but one that can contain a pointer to a range of objects such as associated document files, web pages, Help, model features and even other Enterprise Architect model files. When you double-click on the element, Enterprise Architect executes the link. To add a Hyperlink element, drag the *Hyperlink* icon from the **Common** page of the **Toolbox** onto the diagram.

(Alternatively, click on the **Hyperlink** icon in the **UML Elements** toolbar and then click on the diagram.)



Configure the Hyperlink:

When you add the Hyperlink to the diagram, the **Hyperlink Details** dialog displays. If you want to display the information in a more readable layout, you can resize the dialog.

You first select the type of object to link to; click on the drop-down arrow in the **Type** field and select a type. The **Hyperlink Details** dialog then displays the appropriate fields, prompts or dialog to enable you to specify the object to link to. For example, if you intend to hyperlink to:

- an attribute, the Set Attribute dialog displays to enable you to select that attribute
- a file, the **Action** field displays to enable you to specify whether to **Open** the file in read only mode, or **Edit** the file; in either case the file is opened within Enterprise Architect if possible, or, if not possible, with the Windows default viewer/editor for the file type. For example, if you hyperlink to a .rtf file, you can view it in whichever viewer is appropriate; however, you cannot *edit* .rtf files in Enterprise Architect, so the file always opens in the *Windows* default .rtf editor.
- a diagram, the **Select a Diagram** dialog displays, which enables you to select the diagram from anywhere in the project; you can filter the selection to diagrams of certain types.

In most cases you define the location of the hyperlinked object in the **Address** field, either by overtyping the field or by clicking on the (...) (Browse) button. You can use full paths or local (path substitution) paths.

If you select **EA Command** as the link type, the **Address** field is replaced by a drop-down list of Enterprise Architect commands. You can select **LocalPath** and click on the (...) (Browse) button to display the **Local Paths** dialog, which you complete as required. Subsequently, when you click on the hyperlink the **Local Paths** dialog immediately displays and you can apply, switch, expand or update the current path.

In the **Alias** field, type the text to display in the hyperlink. If you do not provide an alias, either the text defaults to the link itself, or (for certain link targets such as a matrix profile) the dialog generates a simple text instruction.

If you prefer to display only the hyperlink text, without the icon, select the **Hide Icon** checkbox.

Notes:

- If required, you can create a number of empty hyperlinks to complete later; if you then double-click on an empty hyperlink, the **Hyperlink Details** dialog displays and you can enter the details
- Once you have created the hyperlink, you can also edit the hyperlink text by clicking once on the field and once on the text, then right-clicking and selecting the **Edit Selected** context menu option
- You can add notes to the hyperlink, which display in the **Hyperlink Details** dialog when you right-click on the hyperlink and select the **Properties** context menu option; you can format these notes using the **Notes** toolbar

Learn More:

- [Hyperlinks to Files](#)^[1293]
- [Hyperlinks to Scripts](#)^[1294]
- [Add Action as a Hyperlink](#)^[1294]
- [Hyperlinks between Diagrams](#)^[1294]

8.6.5.1 Hyperlinks To Files

To create a hyperlink on a diagram to an external file, simply click on the file in a file list (such as Windows Explorer) or on your Desktop and drag it onto the diagram.

A short context menu displays with two options - **Hyperlink** and **Artifact**. Click on the **Hyperlink** option to create the hyperlink on the diagram.

The link is effective immediately, and you can right-click on it to add or change properties as necessary.

Files of most types - including .sql and .ddl - are opened within the appropriate Enterprise Architect code editor.

Learn More:

- [Hyperlinks](#)^[1292]

8.6.5.2 Script Hyperlinks

You can create a hyperlink on a diagram to execute a script. Simply drag the required script from the [Execution Analyzer](#) ^[1392] window onto the diagram. A context menu displays, from which you select whether the script to be executed is a Build, Test, Run, Debug or Deploy script. The hyperlink is effective immediately; when you click on it, the script executes.

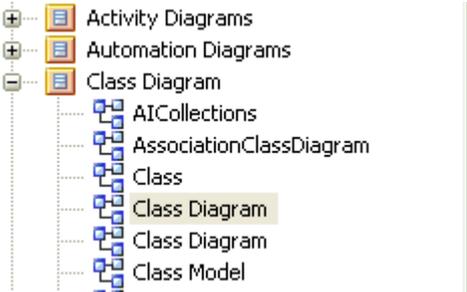
8.6.5.3 Add Action As Hyperlink

To create an Action element to represent a wide range of behaviors and actions, including a hyperlink, follow the steps below

Step	Action
1	Drag an Action element from the Activity page of the Toolbox onto the diagram A context menu immediately displays
2	Select the Other menu option The New Action dialog displays, with the Other radio button selected
3	Click on the drop down arrow on the field in the Select Kind panel, and click on the Hyperlink option
4	Click on the OK button The <i>Hyperlink Action</i> element displays on the diagram
5	Right click on the element and select the Advanced Set Hyperlink menu option The Hyperlink Details dialog displays
6	Set the hyperlink's properties ^[1292]

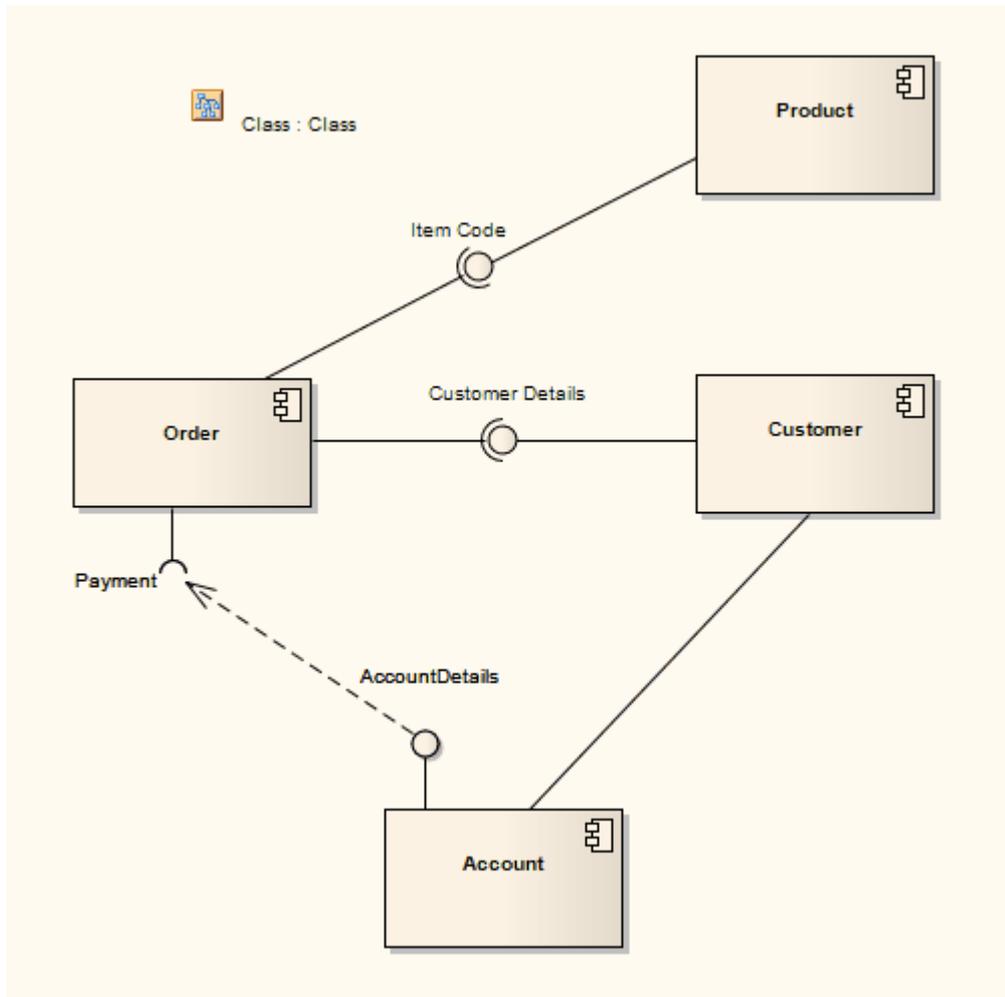
8.6.5.4 Hyperlinks Between Diagrams

To create a hyperlink between diagrams, follow the steps below

Step	Action
1	Open the diagram in which to display the hyperlink to another diagram From the Project Browser select the diagram you want to create a hyperlink to 
2	Drag the diagram on to the current diagram The Select Type dialog displays
3	Select the Hyperlink option and click on the OK button

Step	Action
	The final hyperlinked diagram should resemble the diagram below, where the Class diagram is the diagram to which the Product Order diagram hyperlinks (notice that the hyperlink icon is different)

Example



Notes:

- If the hyperlink appears as a Sub Activity, select the **Tools | Options | Diagram | Behavior** menu option and deselect the **Use Automatic SubActivities** checkbox

8.6.6 Image



Description

An Image is a System Boundary element that automatically displays first the **Boundary Properties** dialog and then the **Select Alternate Image** dialog to change its representation to an imported image. You can use it as an icon for an element or group of elements, or as a diagram background.

Image elements are available from the **Common** page of the **Toolbox**.

Toolbox Icon



Learn More:

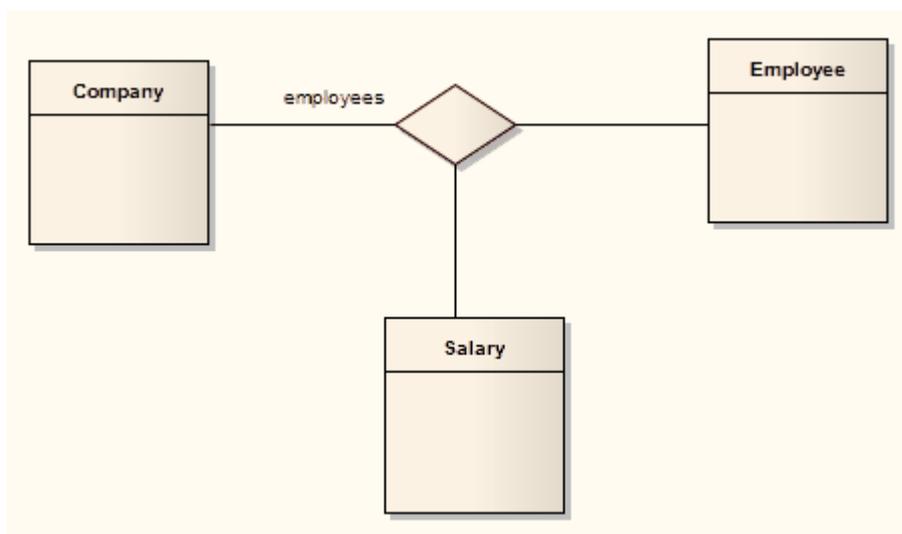
- [System Boundary](#)^[93]

8.6.7 N-Ary Association



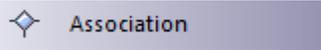
Description

An *n*-Ary Association element is used to model complex relationships between three or more elements, typically in a Class diagram. It is not a commonly-employed device, but can be used to good effect where there is a dependant relationship between several elements. It is generally used with the Association connector, but the relationships can include other types of connector.



In the example above there is a relationship between a *Company*, an *Employee* and a *Salary*.

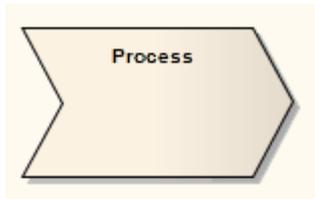
Toolbox Icon



Learn More:

- [Class Diagram](#) ^[798]
- [Association](#) ^[970]

8.6.8 Process



Description:

A Process is an Activity element with the stereotype **process**, which expresses the concept of a business process. Typically this involves inputs, outputs, work flows, goals and connections with other Processes. The Process element is typically used in Analysis diagrams.

Business processes typically range across many parts of the organization and span one or more systems.

Toolbox Icon:



Learn More:

- [Analysis diagram](#) ^[1187]

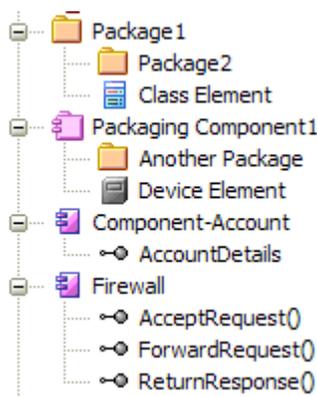
8.6.9 Packaging Component



Description

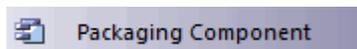
A *Packaging Component* is an element that appears very similar to a Component in a diagram but behaves as a Package in the **Project Browser** (that is, it can be version controlled and can contain other Packages and elements). It is typically used in Component diagrams.

In the **Project Browser**, the three elements display as shown below:



The Component element cannot contain child Packages or Packaging Components.

Toolbox Icon



Learn More:

- [Component diagrams](#)^[807]
- [Component](#)^[948]
- [Package](#)^[960]

8.6.10 Risk



Description

A Risk is defined as the effect of uncertainty on objectives. In Project Management, it is necessary to try to identify risks and assess:

- the likelihood that they have a negative effect on a project and
- how large that effect is likely to be.

Those risks with a high probability of occurrence and/or a large impact on the project can be mitigated.

A Risk Management process might consist of the following five steps:

1. Identify risks and represent each with a Risk element.
2. Identify which elements (e.g. Components, Use Cases, Features) are vulnerable to each risk. You might decide to create «trace» dependencies from these elements to the Risk elements.
3. Assess the likelihood and magnitude of the risks.
4. Identify ways to mitigate the risks.
5. Prioritize the risk reduction measures based on their likelihood, magnitude and ease of mitigation.

Risk *elements* are not the same as the risks that you assign *to* an element through the Project Management

window. Such risks are internal to the selected element, whilst a Risk element can be associated with a number of elements, either in a logical group or totally separate.

Risk elements are available from the Requirements page of the Toolbox.

Notes:

- Risk elements can be displayed with or without an identifying **R** in the top right corner of the element; to toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page.

Toolbox Icon



8.6.11 Task



Description

A Task element represents a task that must be performed in relation to an element. The Task element enables you to assign resources to the task itself, rather than just to the parent element.

You can create a hierarchy or tree structure of Task elements to break a large task into separate parts and assign different resources to each part.

Toolbox Icon



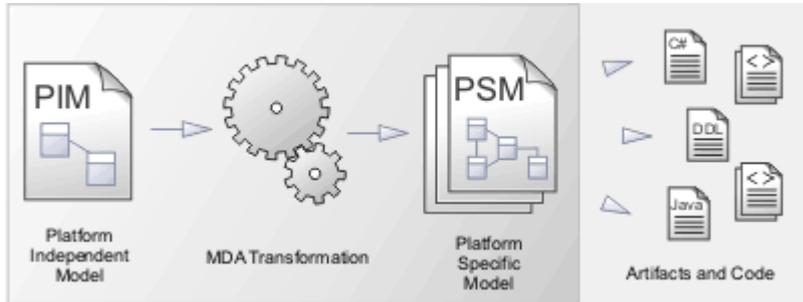
Learn More:

- [Resource Management](#)^[348]
- [Relationship Matrix](#)^[1722]

Part



9 Model Transformation



Model Driven Architecture (MDA) Transformations provide a fully configurable way of converting model elements and model fragments from one domain to another. This typically involves converting Platform-Independent Model (PIM) elements to Platform-Specific Model (PSM) elements. A single element from the PIM can be responsible for creating multiple PSM elements across multiple domains.

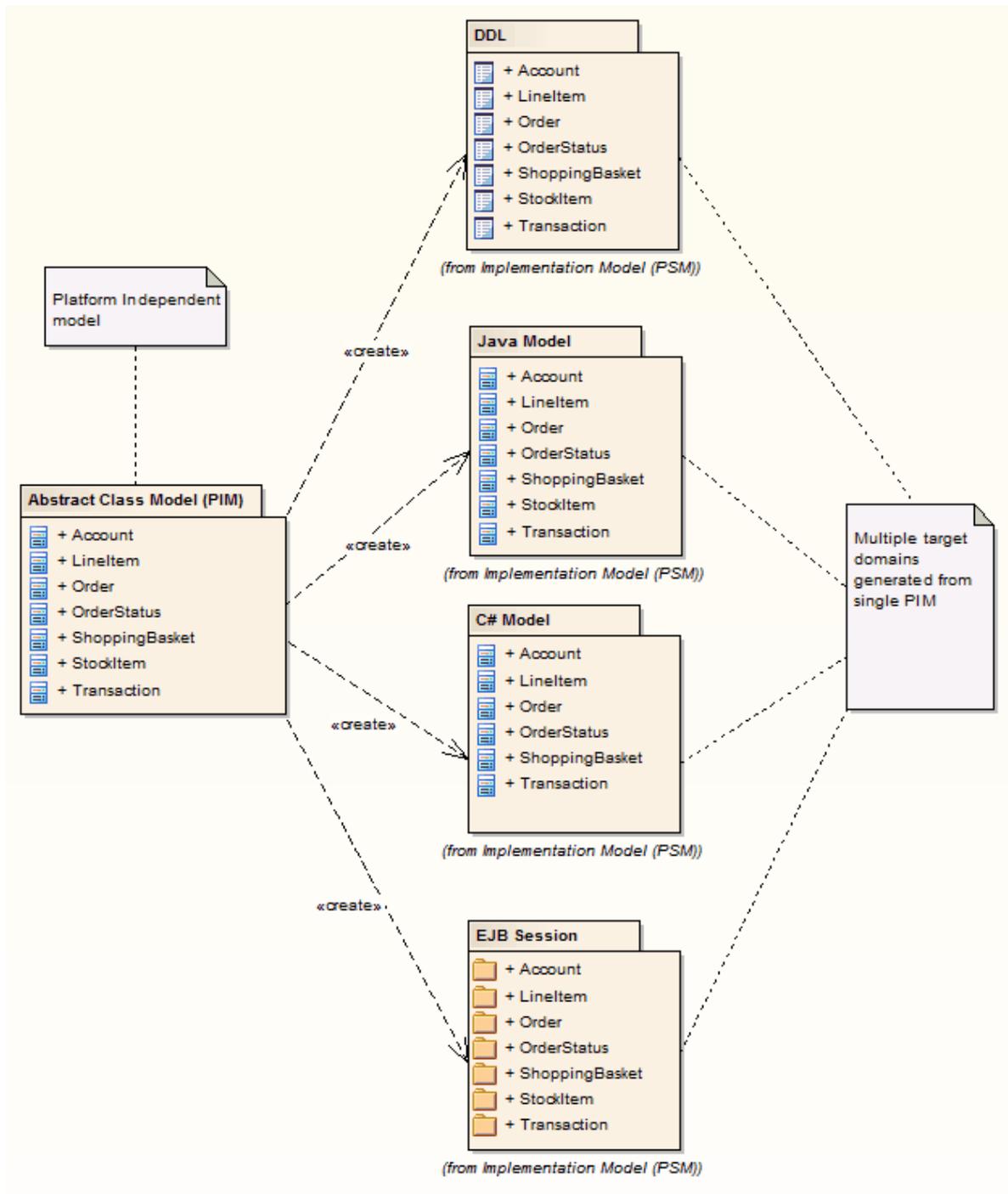
Topics:

Topic	Detail	See also
<p>Overview</p>	<p>Transformations are a huge productivity boost, and reduce the necessity of manually implementing stock Classes and elements for a particular implementation domain; for example, database tables generated from persistent PIM Classes</p> <p>Enterprise Architect includes a number of basic built-in Transformations (see <i>Included Transformations</i>, below); Sparx Systems intend to make further Transformations available over time, either as built in Transformations or as downloadable modules from the Sparx Systems website</p> <p>For a further productivity boost, Enterprise Architect can automatically generate code for your transformed Classes that target code languages; see the Generate Code on result option on the Model Transformation dialog</p> <p>A Transformation is defined using Enterprise Architect's simple code generation template language, and involves no more than writing a template to create a simple intermediary source file; Enterprise Architect reads the source file and binds that to the new PSM</p>	<p>Transform Elements^[1305]</p>
<p>Transformation Dependencies</p>	<p>Enterprise Architect also creates internal bindings (Transformation Dependencies) between each PSM created and the original PIM</p> <p>This is essential, as it enables you to forward synchronize from the PIM to the PSM many times, adding or deleting features as you go; for example, adding a new attribute to a PIM Class can be forward synchronized to a new column in the Data Model</p> <p>You can observe the Transformation Dependencies for a package using the Traceability window; this enables you to check the impact of changes to a PIM element on the corresponding elements in each generated PSM, or to verify where a change required in a PSM should be initiated in the PIM (and also to reflect back in other PSMs)</p>	<p>Traceability^[493]</p>

Topic	Detail	See also
	<p>The Transformation Dependencies are a valuable tool in managing the traceability of your models</p> <p>Enterprise Architect does not delete or overwrite any element features that were not originally generated by the transform; therefore, you can add new methods to your elements, and Enterprise Architect does not act on them during the forward generation process</p>	
<p>Included Transformations</p>	<p>Transformations that are currently built-in include:</p> <ul style="list-style-type: none"> • C# - Converts a PIM to a standard C# implementation set • Data Model to ERD - Transforms a Data Model to an Entity Relationship Diagram (ERD) • DDL - Transforms platform-independent Class elements to platform-specific table elements • EJB Entity - Transforms platform-independent Class elements to packages containing the Class and Interface elements that comprise an EJB Entity Bean • EJB Session - Transforms platform-independent Class elements to packages containing the Class and Interface elements that comprise an EJB Session Bean • ERD to Data Model - Transforms an Entity Relationship Diagram into a Data Model • Java - Transforms platform-independent elements to Java language elements • JUnit - Converts a Java model to a model where test methods are created for each public method of any original Class • NUnit - Converts a .Net language specific model to a model where test methods are created for each public method of any original Class • WSDL - Converts a simple representation of a WSDL interface into the elements required to generate that interface • XSD - Transforms platform-independent elements to XSD elements 	

Example:

The following diagram highlights how Transformations work and how they can significantly boost your productivity:

**Notes:**

- If you are using the Corporate, Business and Software Engineering, System Engineering or Ultimate edition, if security is enabled you must have **Transform Package** access permission to perform an MDA Transform on a package

Learn More:

- [Permission List](#)^[205]

- [Transform Elements](#) [1305]
- [Import Transformations](#) [1331]
- [Transformation Templates](#) [1330]
- [Built-in Transformations](#) [1307]
- [Write Transformations](#) [1332]
- [Chaining Transformations](#) [1306]

9.1 Transform Elements

There are two modes for initiating a Model Transformation, each of which can be started in two ways.

Access:

To transform selected elements on a diagram: **Tools | Model Transformation (MDA) | Transform Selected Elements**

Class context menu | Transform (Ctrl + H)

To transform elements in the package currently selected in the Project Browser:

Package **Tools | Model Transformation (MDA) | Transform Current**

Current Package **Project Browser package context menu | Transform**

Use To:

- Transform elements
- Transform packages

Reference:

Field	Usage	See also
Elements	Indicates the individual elements to be included in the transformation	
All	Indicates all of the elements from the list to be included in the transformation	
None	Deselects all of the elements from the list	
Include child packages	Specifies inclusion of elements in child packages of the selected package	
Transformations	Indicates which transformations to perform and the package each of them should be transformed to (use the (...) button to select the package in which the transformed elements are being created)	
Generate Code on result	Specify whether or not to automatically generate code for transformed Classes that target code languages Automatically generating code helps boost productivity in development With this option selected, the first time you transform to the selected Class Enterprise Architect enables you to select a filename to generate to; subsequent transformations automatically generate any Class with a filename set	
Perform Transformations on result	Specify if transformations previously done on target Classes should be automatically executed	Chaining Transformations <small>[1306]</small>
Intermediary File	Specify the filename of the intermediary file (if any)	

Field	Usage	See also
Write Always	Write the intermediary file to disk	
Write Now	Generate the intermediary file but do not perform the transform	
Do Transform	Execute the transform command	

Notes:

- When the dialog displays, all elements are selected and all transformations previously performed from any of these Classes are checked

9.1.1 Chaining Transformations

Chaining transformations provide an extra degree of flexibility and power to transformations.

Topics:

Topic	Detail	See also
Example	Two transformations have a common element; you might separate this element out into its own transformation, and then perform the original transformations from the common point The separated transform could even produce a useful model itself	
Setting	Enterprise Architect provides for chaining transformations, by enabling you to set transformations that have already been performed on target Classes to be performed automatically next time that Class is transformed to To do this, select the Perform Transformations on result checkbox in the Model Transformation dialog	

9.2 Built-in Transformations

Enterprise Architect provides a number of built-in transformations. These transformations are intended to be useful to as many users as possible, acting as a good base to modify to include the specifics of your custom domain, and to be good examples of how to write transformations.

The following transformations are included in Enterprise Architect:

Transformations	Detail	See also
C#	Converts Platform-Independent Model (PIM) elements to language-specific C# Class elements	C# Transformation <small>[1307]</small>
Data Definition Language	Creates a data model from the logical model, generating a model targeted at the default database type that is ready for DDL generation	DDL Transformation <small>[1310]</small>
Entity Relationship Diagram to Data Model	Creates a data model from the ERD logical model, generating a model targeted at the default database type ready for DDL generation	ERD To Data Model Transformation <small>[1317]</small>
Data Model to Entity Relationship Diagram	A reverse of the ERD to Data Model transformation	Data Model To ERD Transformation <small>[1310]</small>
EJB Session Bean	Generates a single Class elements to EJB session	EJB Transformations <small>[1314]</small>
EJB Entity Bean	Generates a single Class elements to EJB entity	EJB Transformations <small>[1314]</small>
Java	Converts Platform-Independent Model (PIM) elements to language-specific Java Class elements	Java Transformation <small>[1319]</small>
JUnit	Creates a Class with test methods for all public methods of an existing Java Class	JUnit Transformation <small>[1321]</small>
NUnit	Creates a Class with test methods for all public methods of an existing .Net compatible Class	NUnit Transformation <small>[1323]</small>
WSDL	Creates an expanded model of a WSDL interface from a simple model, which is suitable for generation	WSDL Transformation <small>[1325]</small>
XSD	Converts Platform-Independent Model (PIM) elements to UML Profile for XML elements as an intermediary step to creating an XML Schema	XSD Transformation <small>[1326]</small>

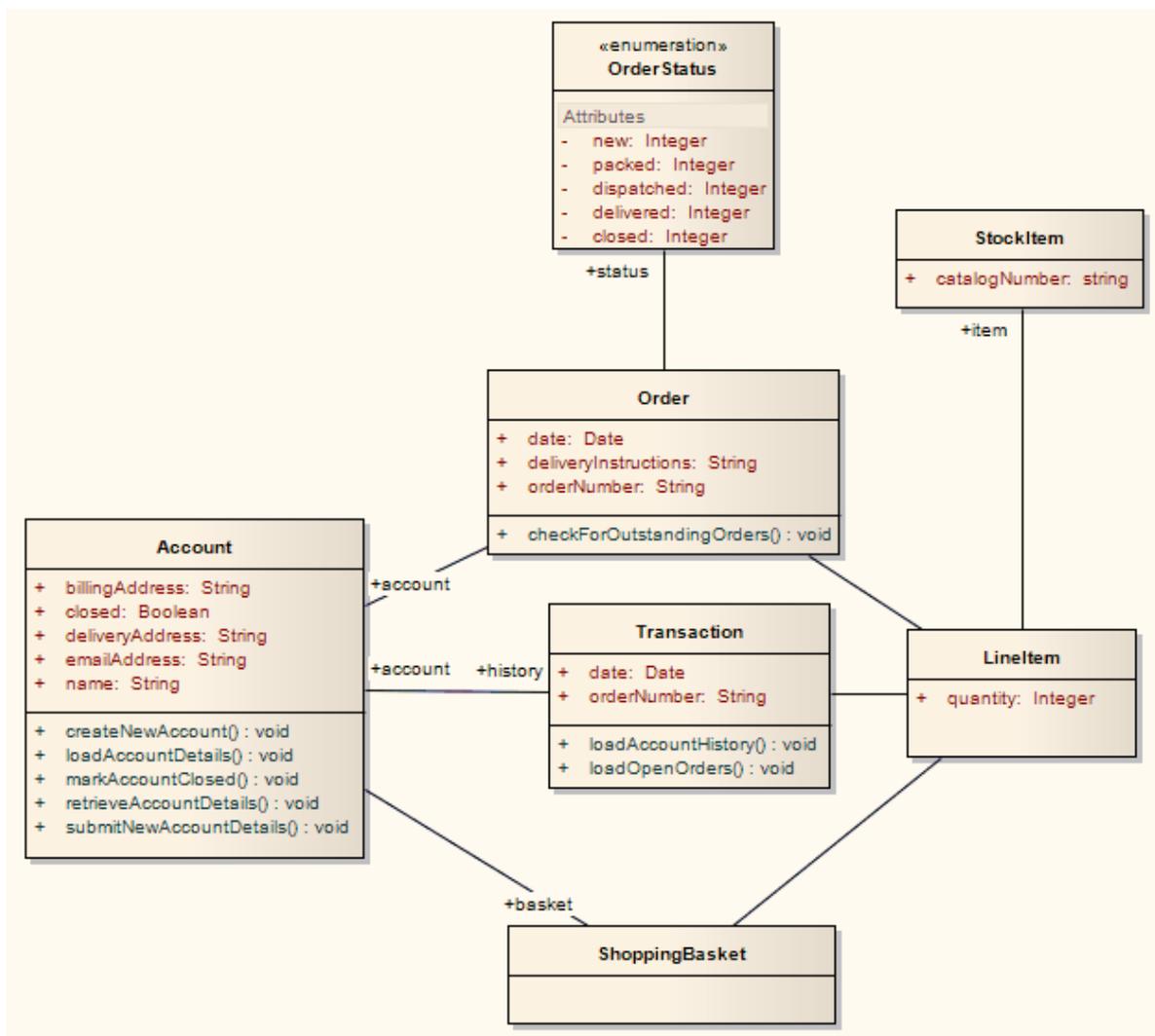
9.2.1 C# Transformation

The **C# transformation** converts Platform-Independent Model (PIM) elements to language-specific C# Class elements.

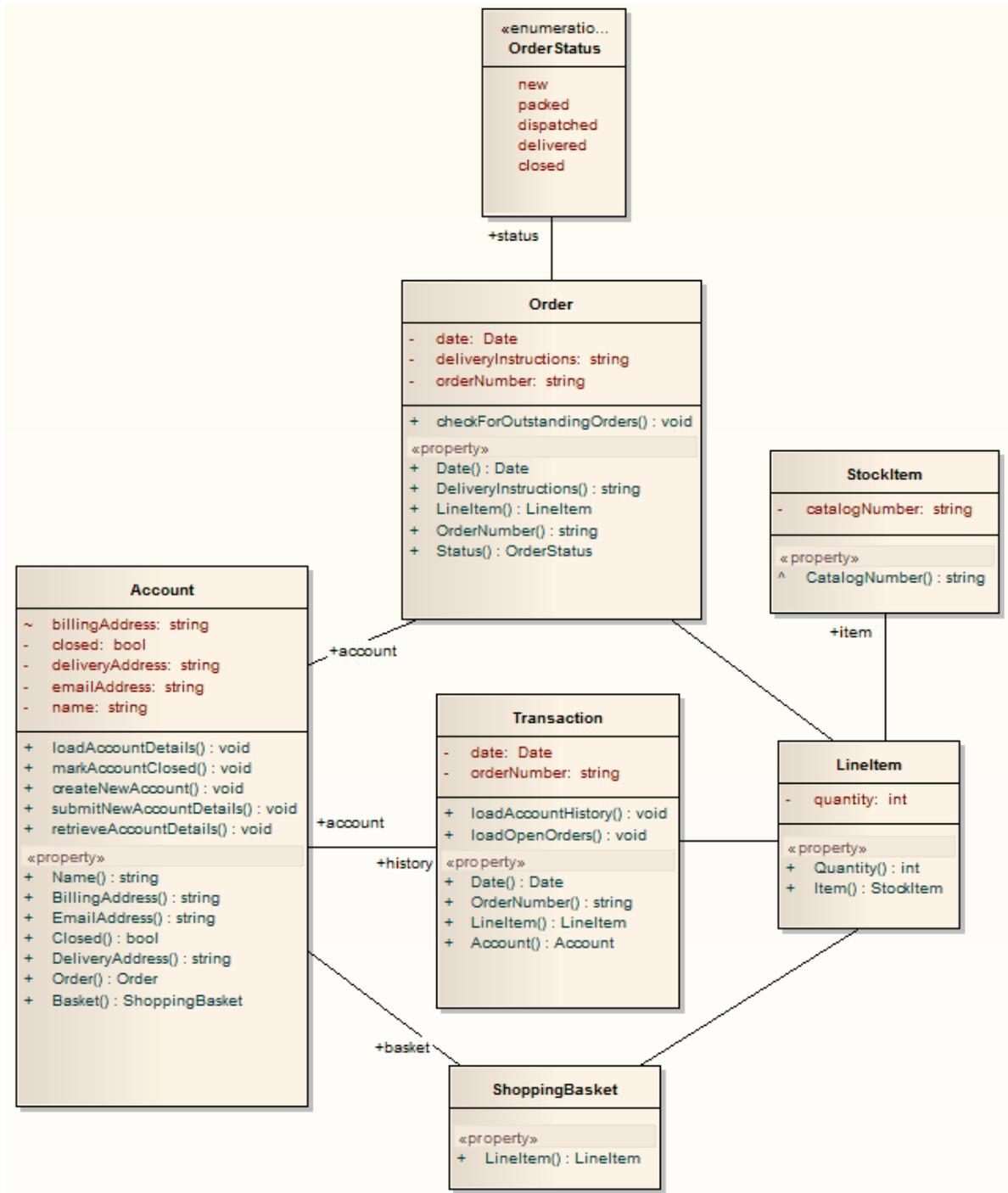
The transformation converts PIM model types to C# types and creates encapsulation according to Enterprise Architect's options for creating properties from C# attributes, which you set on the C# Specifications page of the Options dialog.

Example:

The Platform-Independent Model (PIM):



After transformation, becomes the PSM:



Learn More:

- [C# Options Page](#)¹⁵³²

9.2.2 Data Model To ERD Transformation

The **Data Model to ERD transformation** is a reverse engineering of the **ERD to Data Model transformation**.

The transformation uses and demonstrates support in the intermediary language for the following database-specific concepts:

Reference:

Concept	Definition	See also
Entity	Mapped one-to-one onto table elements.	
Attribute	Mapped one-to-one onto columns.	
Primary Key	Comes from the PrimaryKey type of column.	

Notes:

- Sometimes you might want to limit the stretch of the diamond-shape Relationship connectors. Simply pick a Relationship connector, right-click to display the context menu, and select the **Bend Line at Cursor** option

Learn More:

- [ERD to Data Model Transformation](#)^[1317]

9.2.3 DDL Transformation

The purpose of the **DDL transformation** is to create a data model from the logical model, generating a model targeted at the default database type that is ready for DDL generation. The data model can then be used to automatically generate DDL statements to run in one of the Enterprise Architect supported database products.

The **DDL Transformation** uses and demonstrates support in the **intermediary language** for the following database-specific concepts:

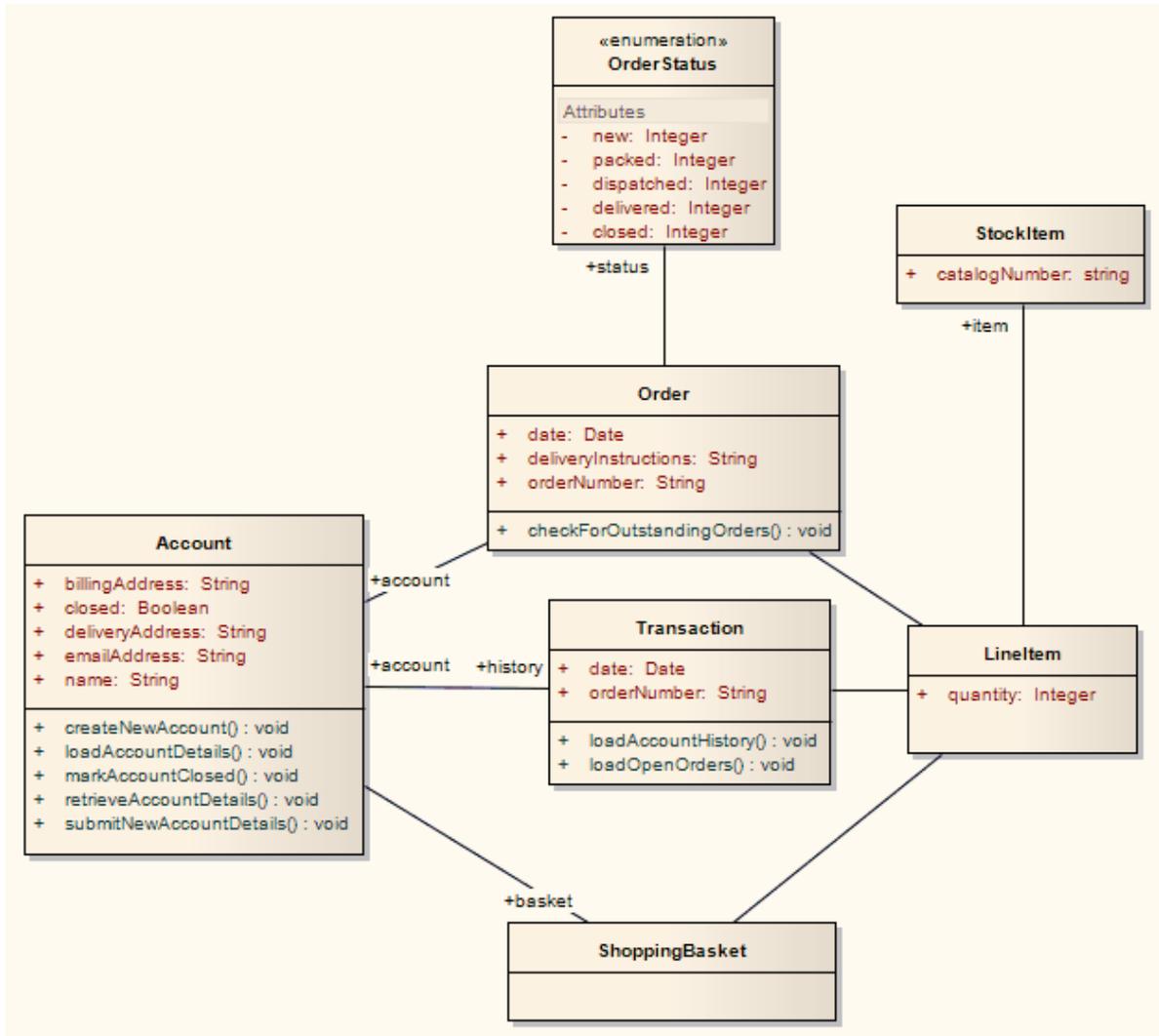
Reference:

Concept	Definition	See also
Table	Mapped one-to-one onto Class elements.	
Column	Mapped one-to-one onto attributes.	
Primary Key	Lists all the columns involved; this ensures that they exist in the Class and creates a primary key method for them.	
Foreign Key	This is a special sort of connector. The <i>Source</i> and <i>Target</i> sections list all of the columns involved; this ensures that they exist and that a matching primary key exists in the destination Class, and that the transformation creates the appropriate foreign key.	

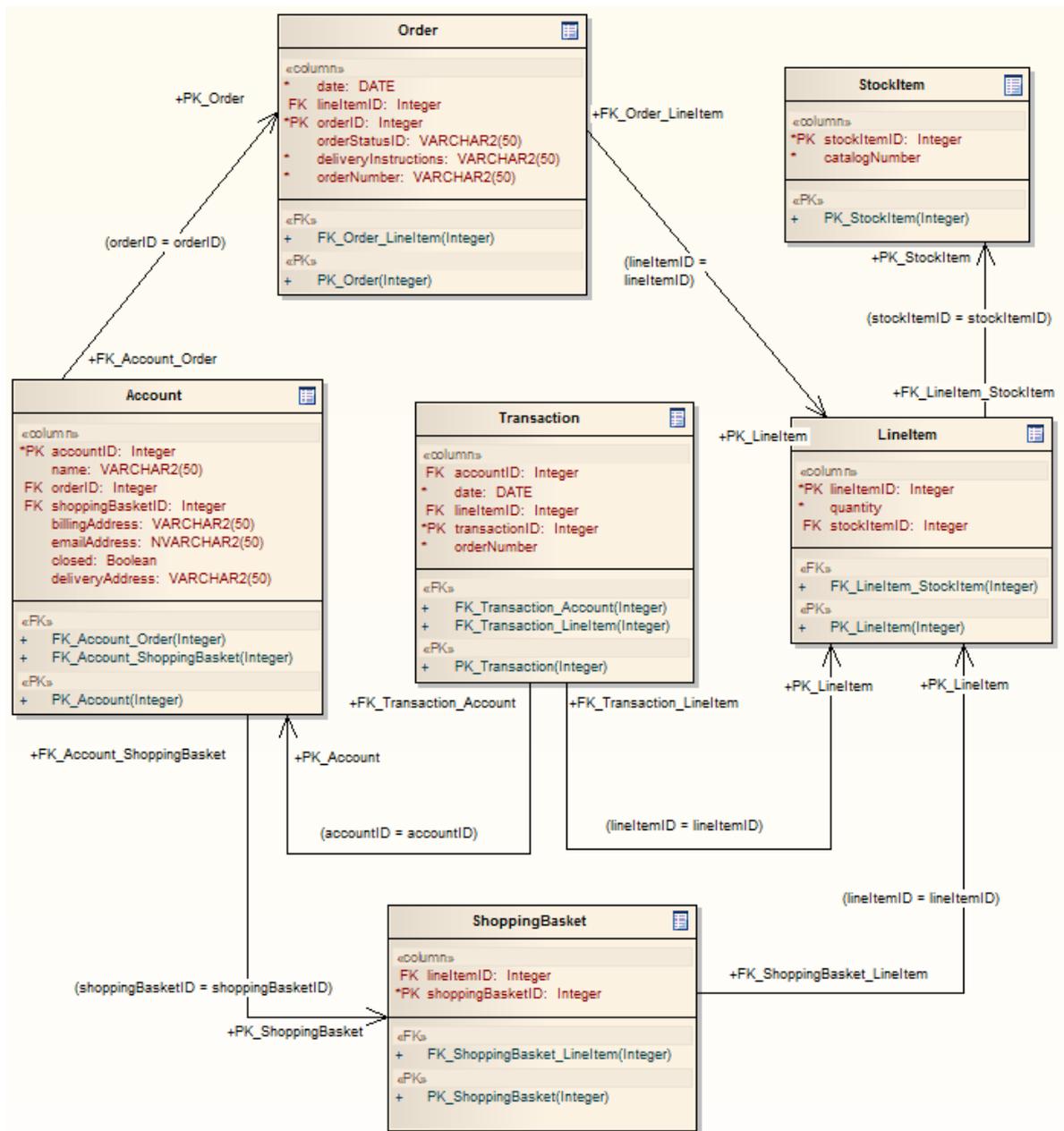
Example:

The following two diagrams show a typical PIM to Data Model Transformation.

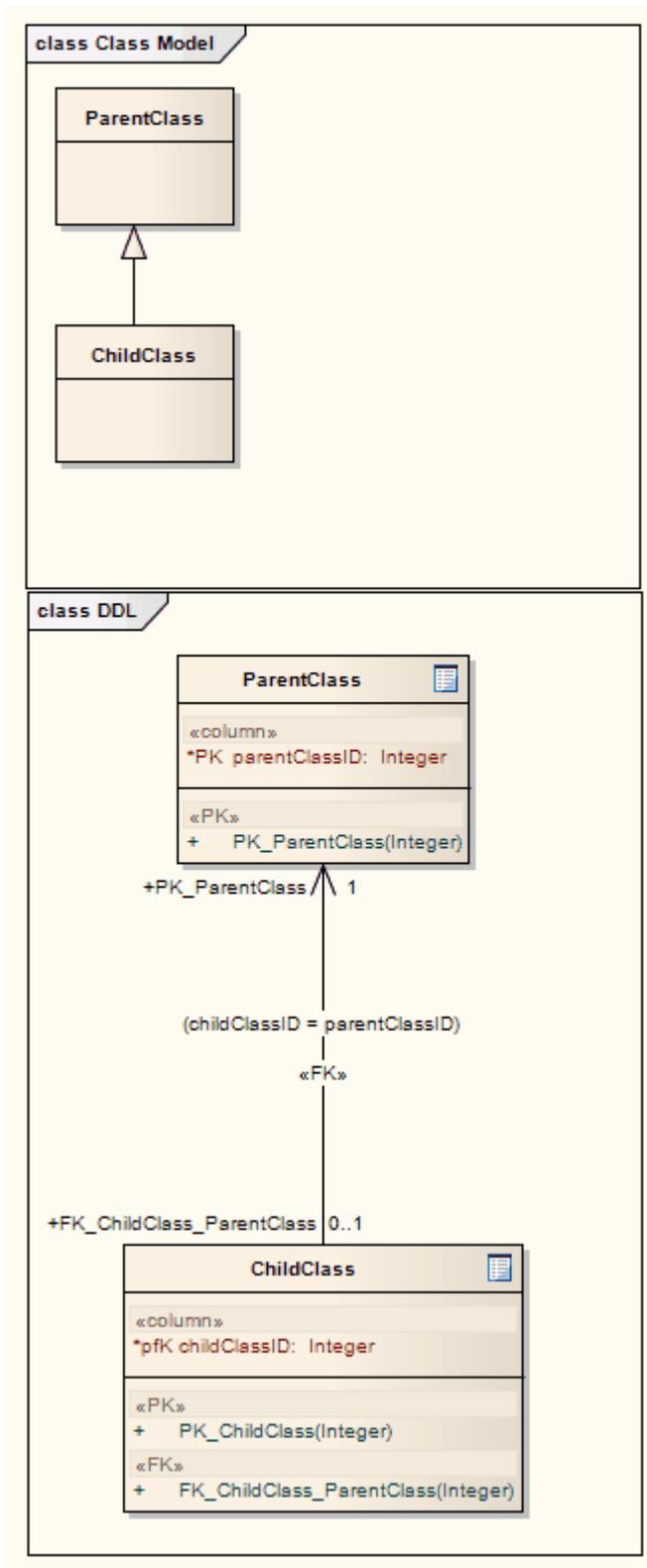
The Platform-Independent Model (PIM):



After transformation becomes the PSM:



Generalizations are handled by providing the child element with a foreign key to the parent element, as in the following diagram. Copy-down inheritance is not supported.



Learn More:

- [Intermediary Language](#) I1333

9.2.4 EJB Transformations

The purpose of the **EJB Session Bean transformation** and the **EJB Entity Bean transformation** is to reduce the work required in generating the internals of Enterprise Java Beans, thus enabling you to concentrate on modeling at a higher level of abstraction.

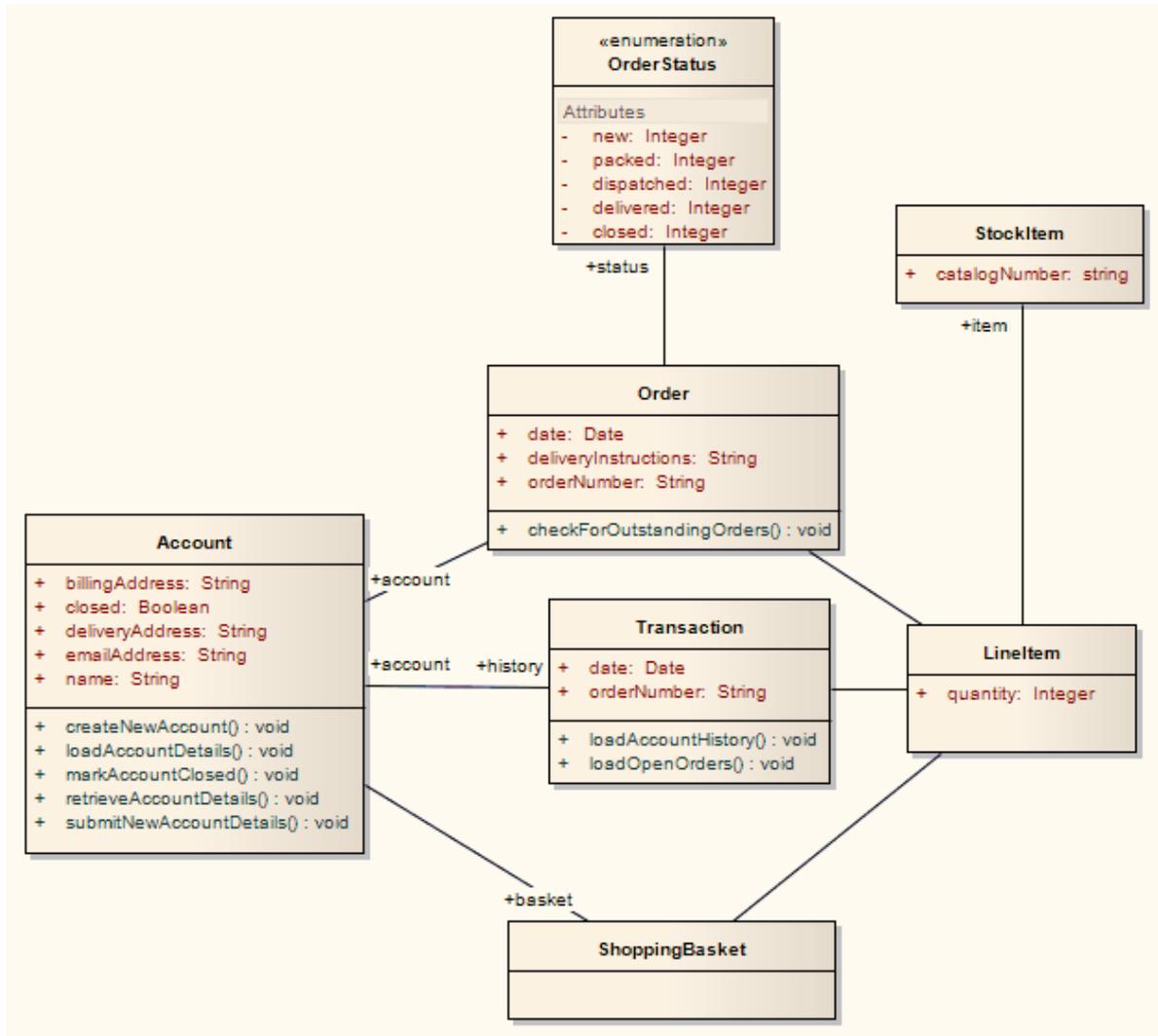
Topics:

Topic	Detail	See also
EJB Session Bean transformation	The EJB Session Bean transformation generates the following from a single Class element containing the attributes, operations and references required for code generation by the <code>javax.ejb.*</code> package: <ul style="list-style-type: none">• An implementation Class element• A home interface element• A remote interface element	
EJB Entity Bean transformation	The EJB Entity Bean transformation generates the following from a single Class element containing the attributes, operations and references required for code generation by the <code>javax.ejb.*</code> package: <ul style="list-style-type: none">• An implementation Class element• A home interface element• A remote interface element• A primary key element	

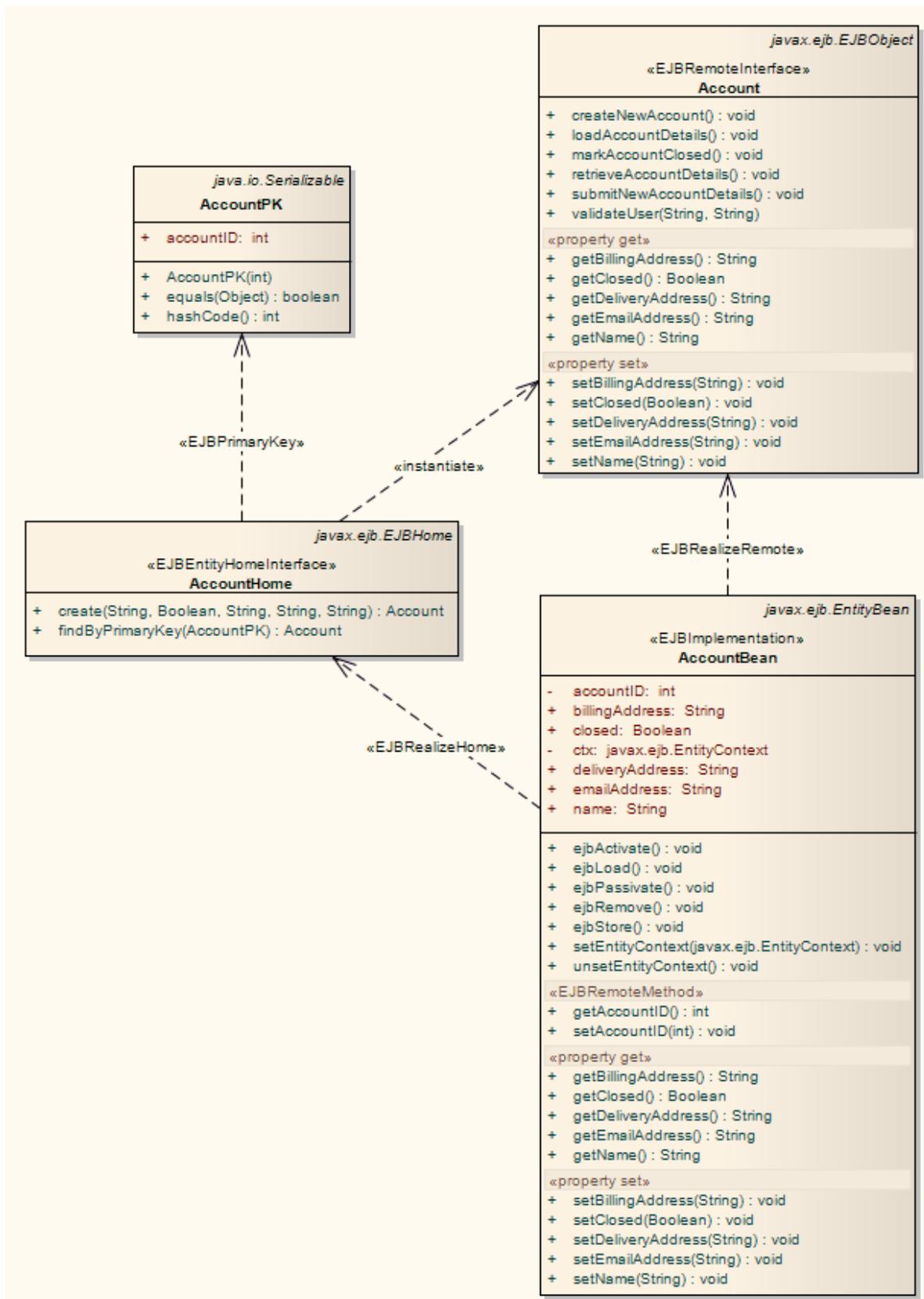
Both transformations also generate a META-INF package containing a deployment descriptor element.

Example:

The Platform-Independent Model (PIM):



After transformation generates a set of Entity Beans, where each one takes the following form (for the Account Class):



9.2.5 ERD To Data Model Transformation

The purpose of the **Entity Relationship Diagram (ERD) to Data Model transformation** is to create a data model from the ERD logical model, generating a model targeted at the default database type ready for DDL generation. Before doing the transformation, make sure you have defined the common data type for each attribute and selected a database type as the default database. The data modeling diagram can then be automatically generated. This data model can be used for generating DDL statements to run in one of the Enterprise Architect supported database products.

The transformation uses and demonstrates support in the intermediary language for the following database-specific concepts:

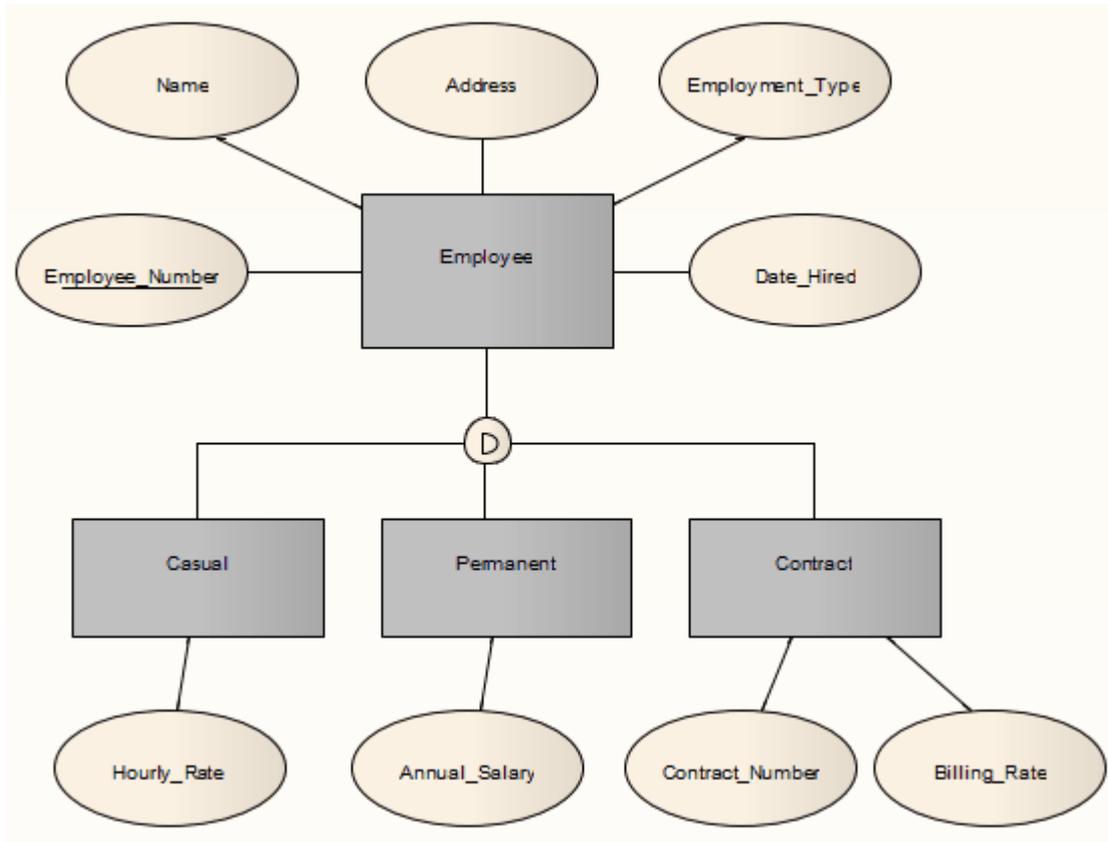
Reference:

Concept	Definition	See also
Table	Mapped one-to-one onto Class elements.	
Column	Mapped one-to-one onto attributes.	
Primary Key	Lists all the columns involved; this ensures that they exist in the Class and creates a primary key method for them.	
Foreign Key	This is a special sort of connector. The <i>Source</i> and <i>Target</i> sections list all of the columns involved; this ensures that they exist and that a matching primary key exists in the destination Class, and that the transformation creates the appropriate foreign key.	

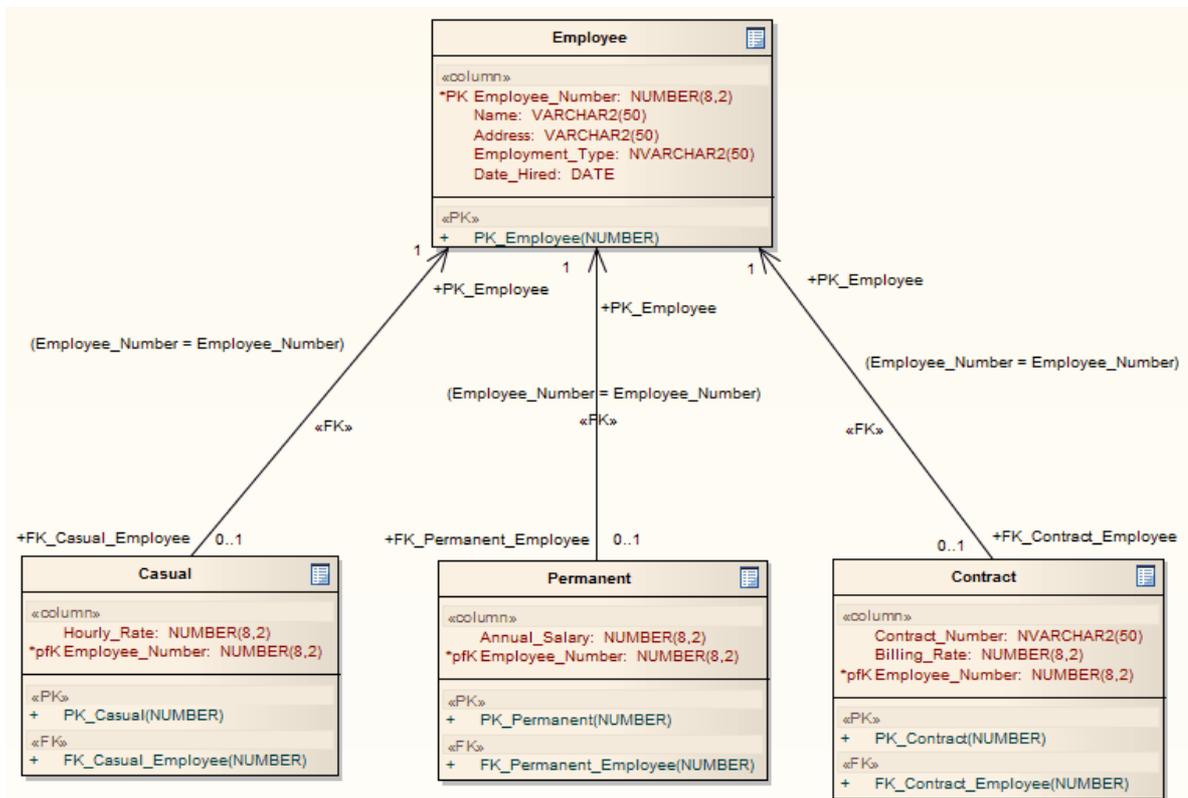
Generalization

Generalization can be handled in ERD technology, as illustrated by the following example. Note that the copy-down inheritance is currently supported with two levels only.

Example:



This transforms to:



Notes:

- Sometimes you might go back to the ERD, make some changes and then want to do another transformation. To achieve better results, always delete the previous transformation package before doing the next transformation

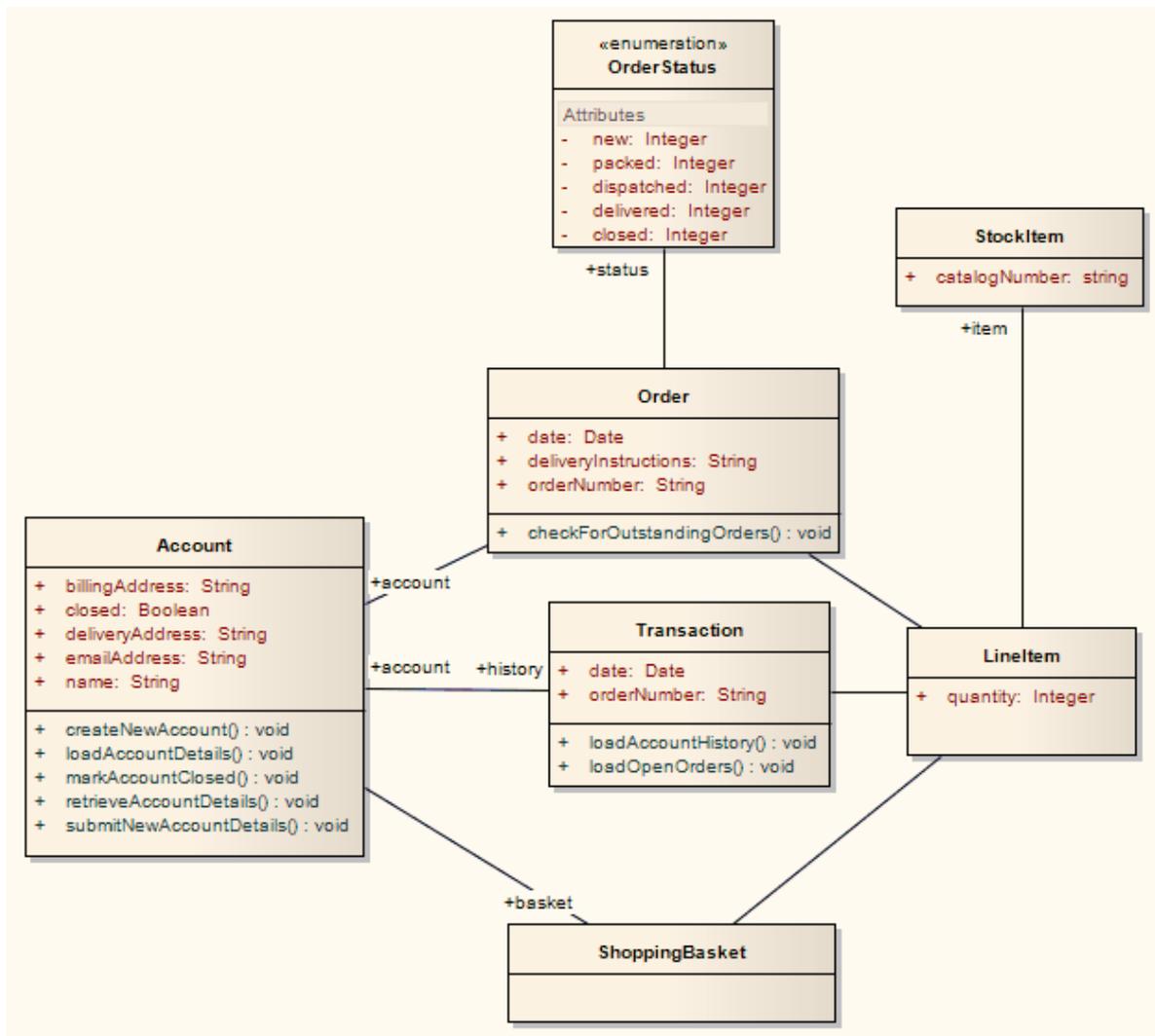
9.2.6 Java Transformation

The purpose of the **Java transformation** is to convert Platform-Independent Model (PIM) elements to language-specific Java Class elements. The transformation converts the PIM model types to Java types and creates encapsulation according to Enterprise Architect's options for creating properties from Java attributes; that is, producing the getters and setters according to the rules you have defined. Notice that the *public* attributes in the PIM are converted to *private* attributes in the PSM.

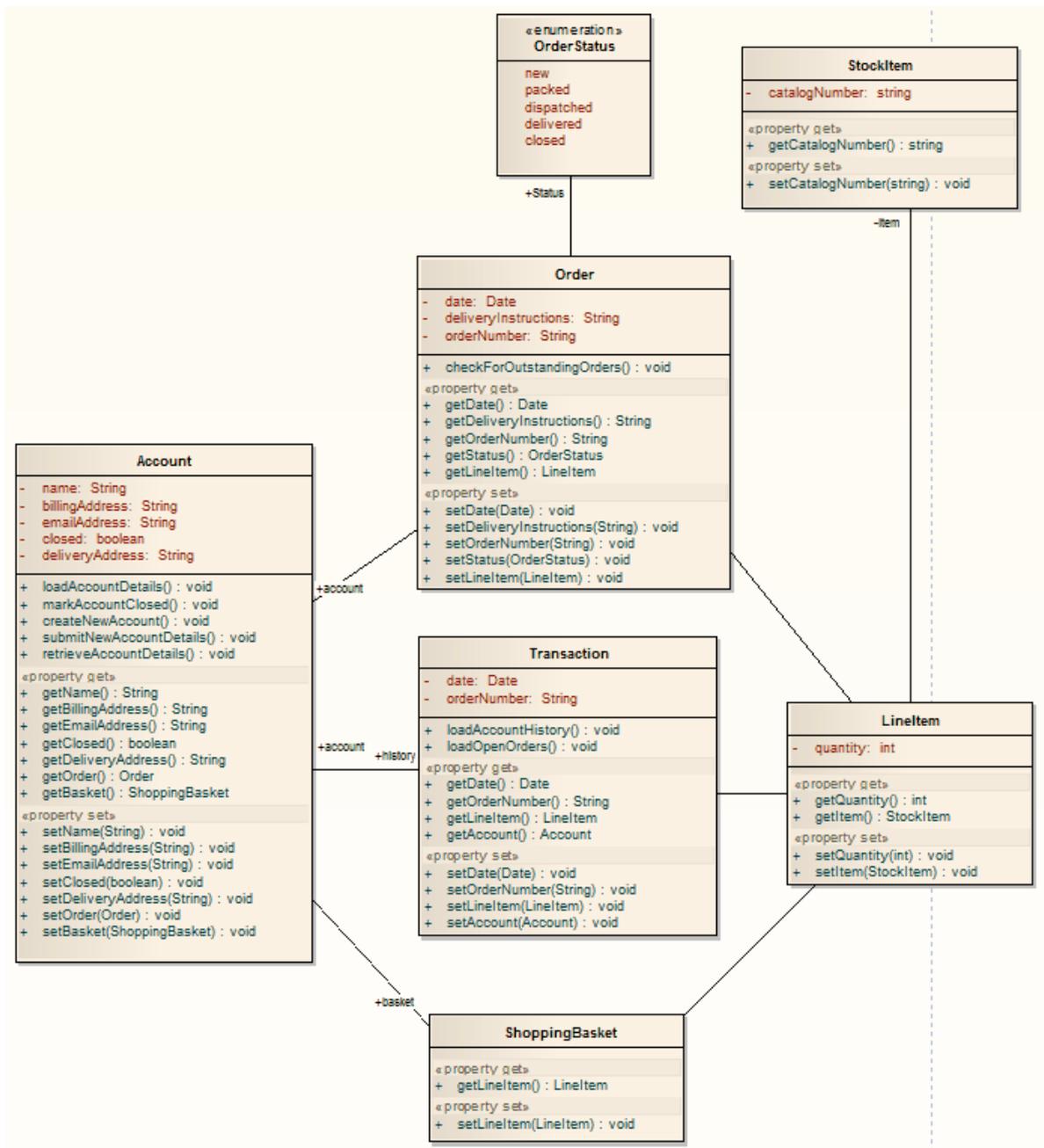
You set the code generation options for Java code generation on the **Java Specifications** page of the **Options** dialog.

Example:

The Platform-Independent Model (PIM):



After transformation becomes the PSM:



Learn More:

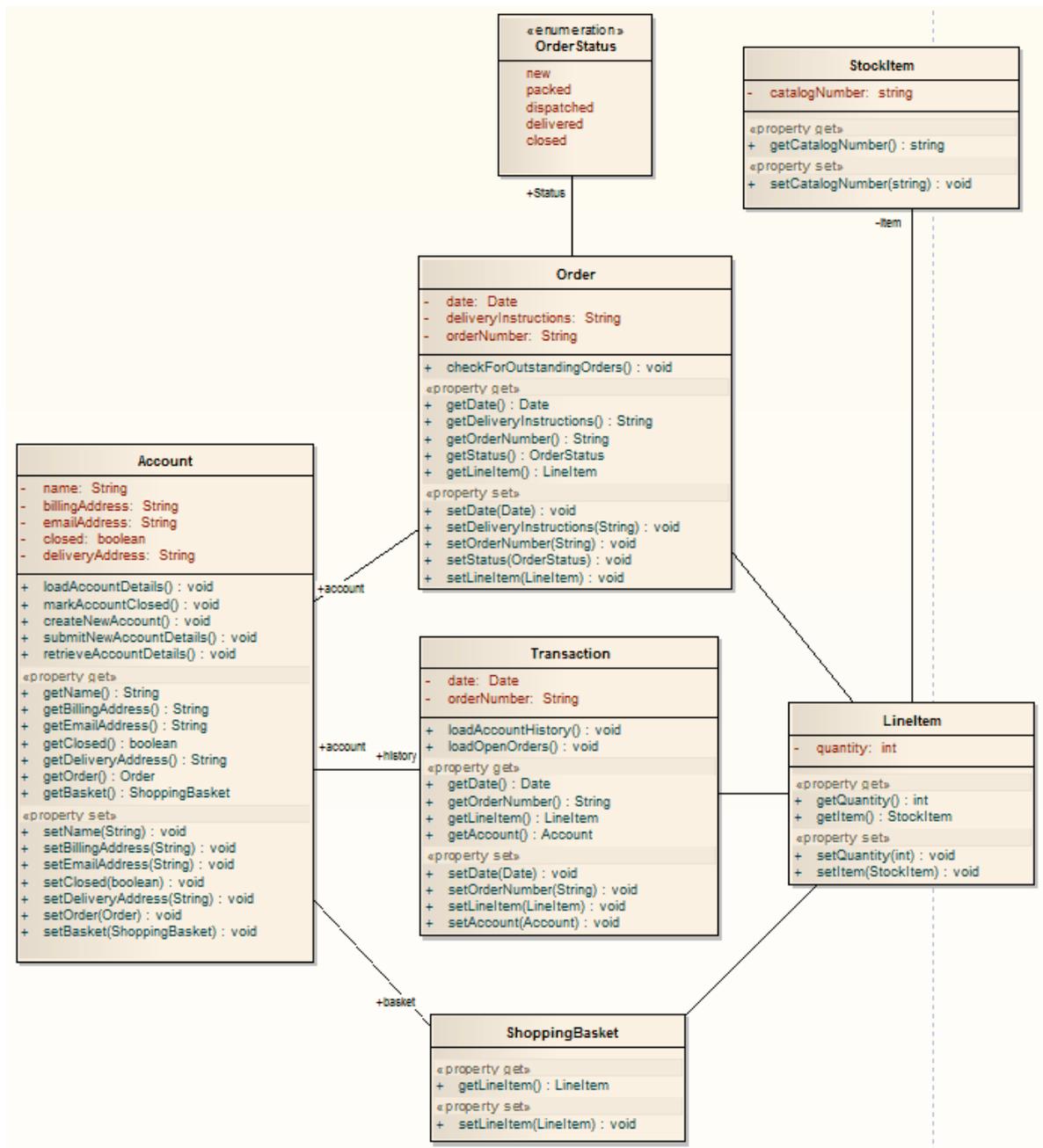
- [Java Options](#) 

9.2.7 JUnit Transformation

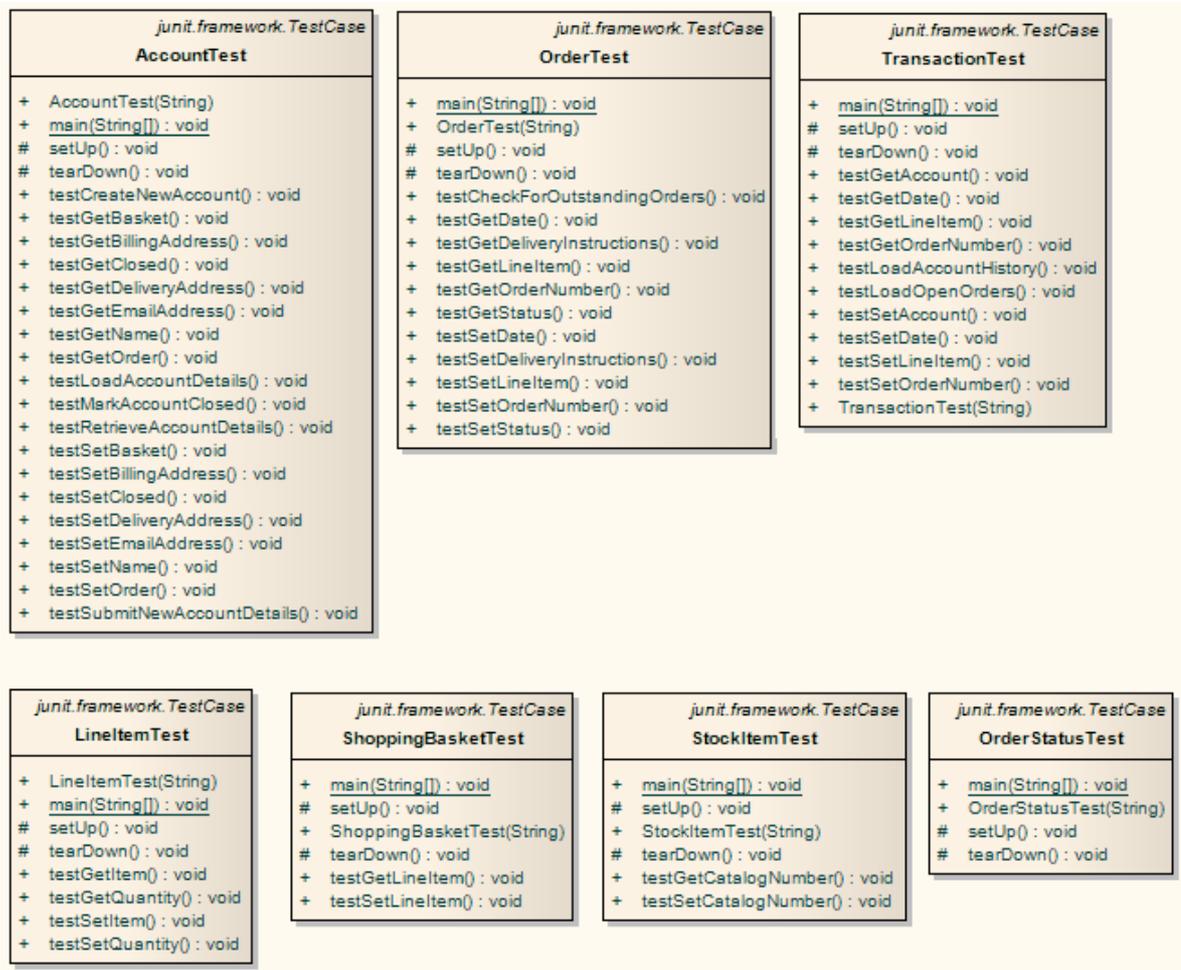
The purpose of the **JUnit transformation** is to create a Class with test methods for all public methods of an existing Java Class. The resulting Class can then be generated and the tests filled out and run by JUnit.

Example:

The Java model originally transformed from the PIM:



After transformation becomes the PSM:



Notes:

- Note that for each Class in the Java model, a corresponding test Class has been created. Each of these test Classes contains a test method for every public method in the source Class, plus the methods required to appropriately set up the tests. It is your responsibility to fill in the details of each test

Learn More:

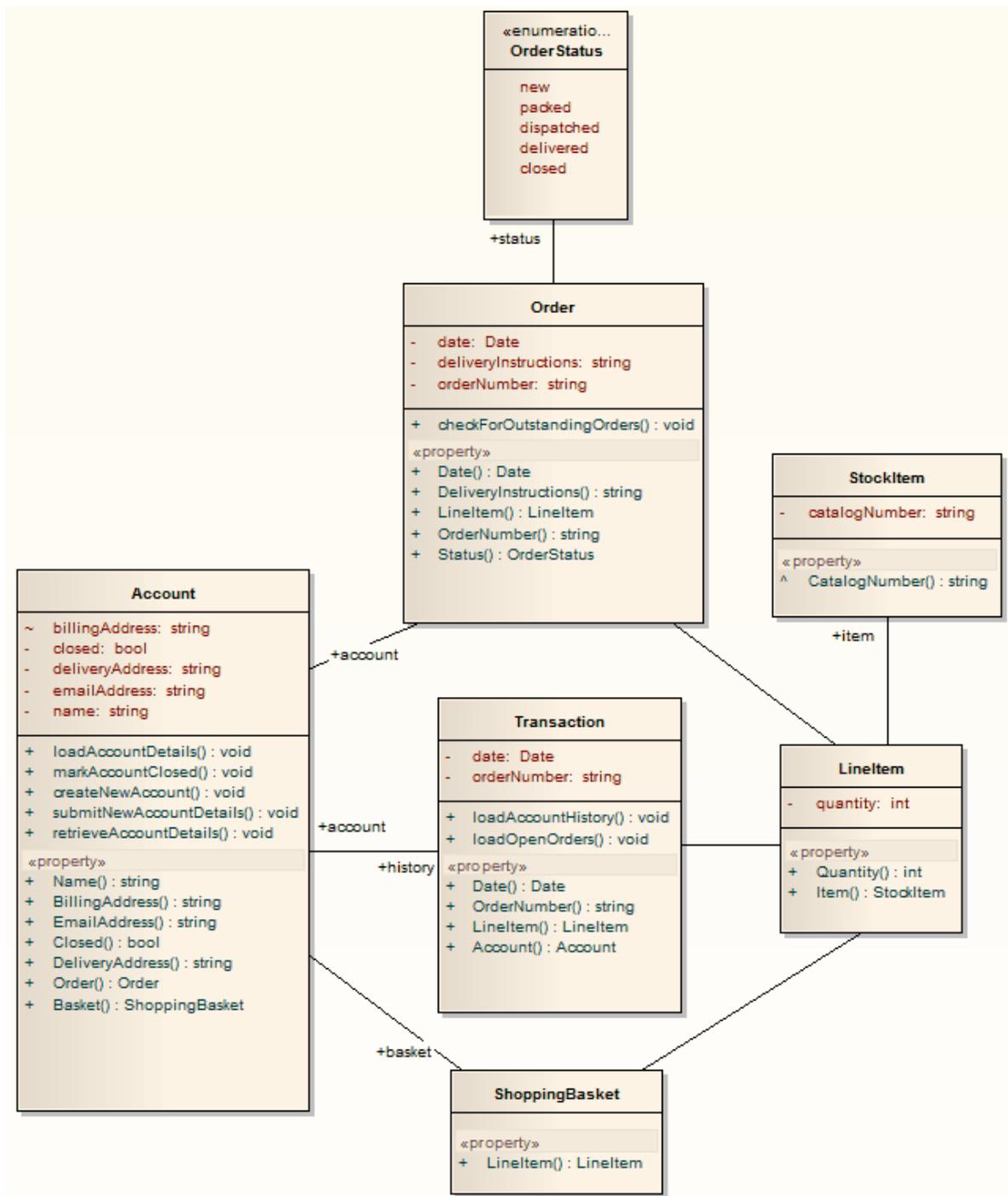
- [Unit Testing](#)^[1669]

9.2.8 NUnit Transformation

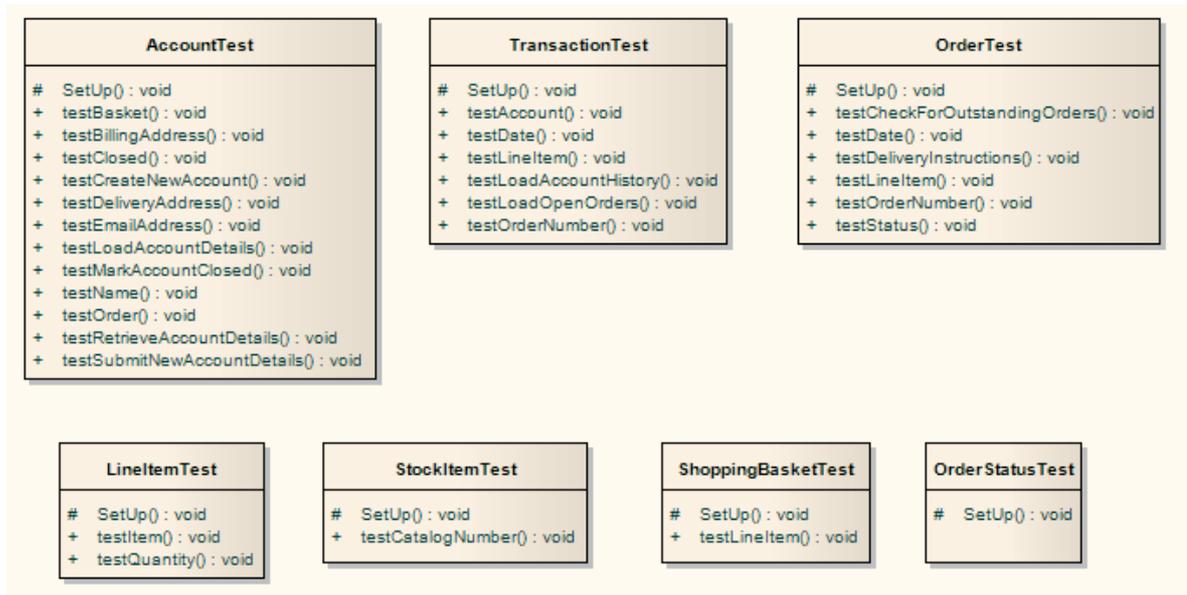
The purpose of the **NUnit transformation** is to create a Class with test methods for all public methods of an existing .Net compatible Class. The resulting Class can then be generated and the tests filled out and run by NUnit.

Example:

The C# model originally transformed from the PIM:



After transformation becomes the PSM:



Notes:

- Note that for each Class in the C# model, a corresponding test Class has been created. Each of these test Classes contains a test method for every public method in the source Class, plus the methods required to appropriately set up the tests. It is your responsibility to fill in the details of each test

Learn More:

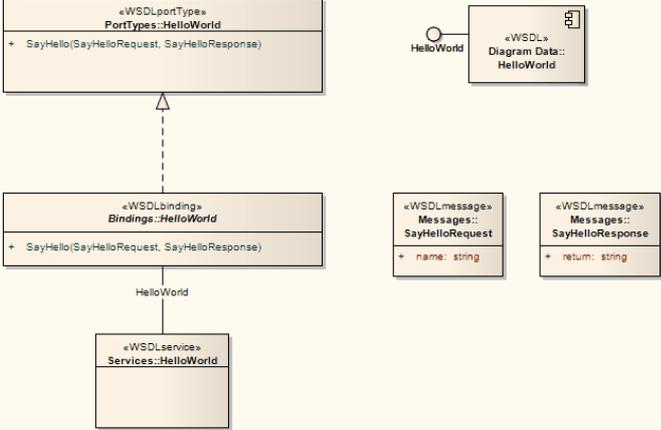
- [Unit Testing](#)^[1669]

9.2.9 WSDL Transformation

The purpose of the **WSDL transformation** is to create from a simple model an expanded model of a WSDL interface that is suitable for generation.

Topics:

Topic	Detail	See also
WSDL Transformation	<p>Take the following example interface:</p> <pre> classDiagram class HelloWorld { <<interface>> + SayHello(name :string) : string } </pre> <p>This generates the corresponding WSDL component, service, port type, binding and messages as follows.</p> <ul style="list-style-type: none"> • Classes are handled in the same way as the XSD 	<p>XSD Transformation^[1326]</p> <p>WSDL Generation^[1621]</p>

Topic	Detail	See also
	<p>Transformation</p> <ul style="list-style-type: none"> All <i>in</i> parameters are transformed into WSDL <i>Message Parts</i> in the <i>Request</i> message The return value and all <i>out</i> and <i>return</i> parameters are transformed into WSDL <i>Message Parts</i> in the <i>Response</i> message All methods where a value is returned are transformed into <i>Request-Response</i> operations while all methods not returning a value are transformed into <i>OneWay</i> operations The transformation does not handle generation of <i>Solicit-Response</i> and <i>Notification</i> methods or faults  <p>The resulting package can then have the specifics filled out using the WSDL editing capabilities of Enterprise Architect, and finally be generated using WSDL generation.</p>	

Learn More:

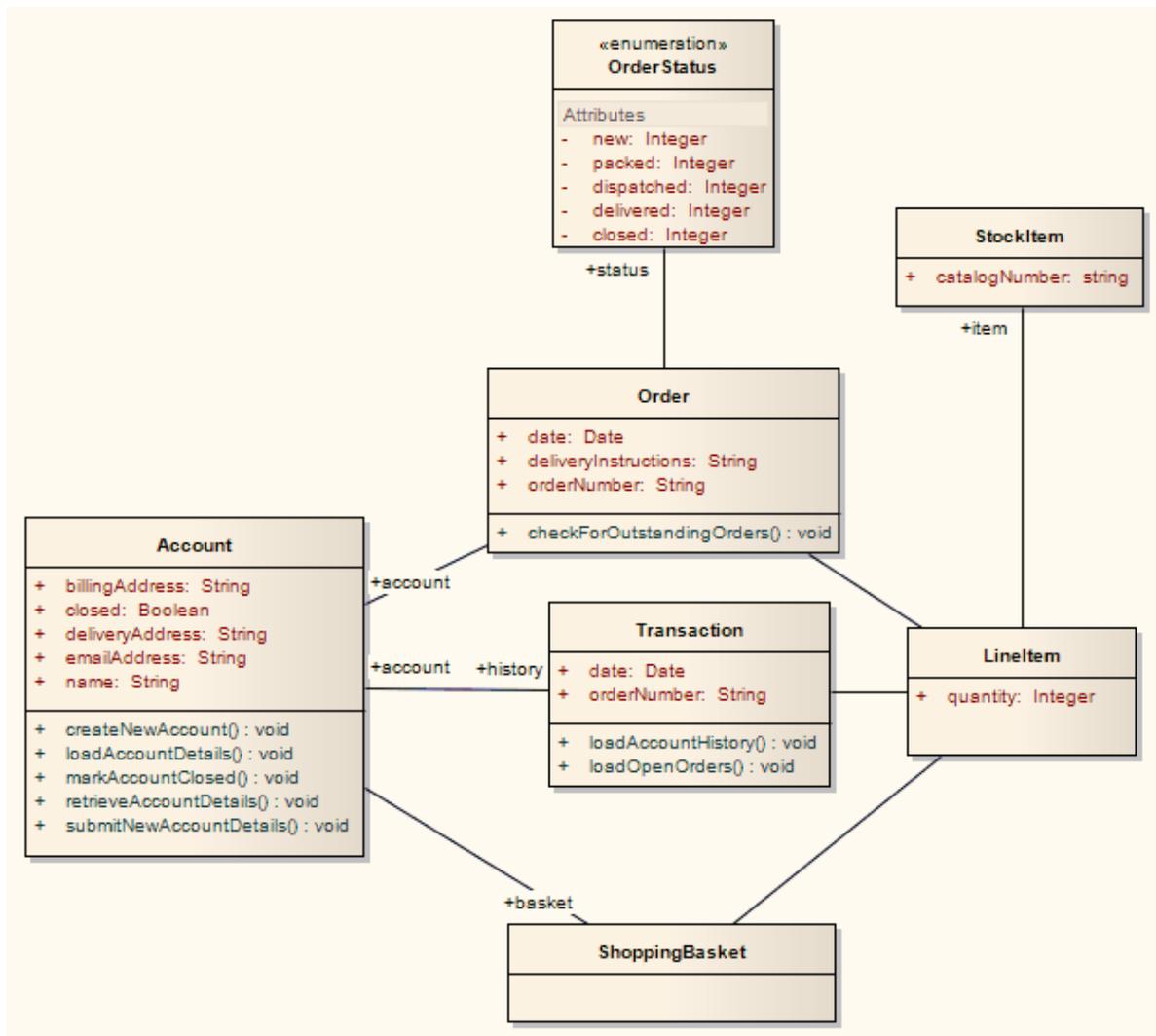
- [WSDL Model Structure](#) ^[1611]
- [Model WSDL](#) ^[1612]

9.2.10 XSD Transformation

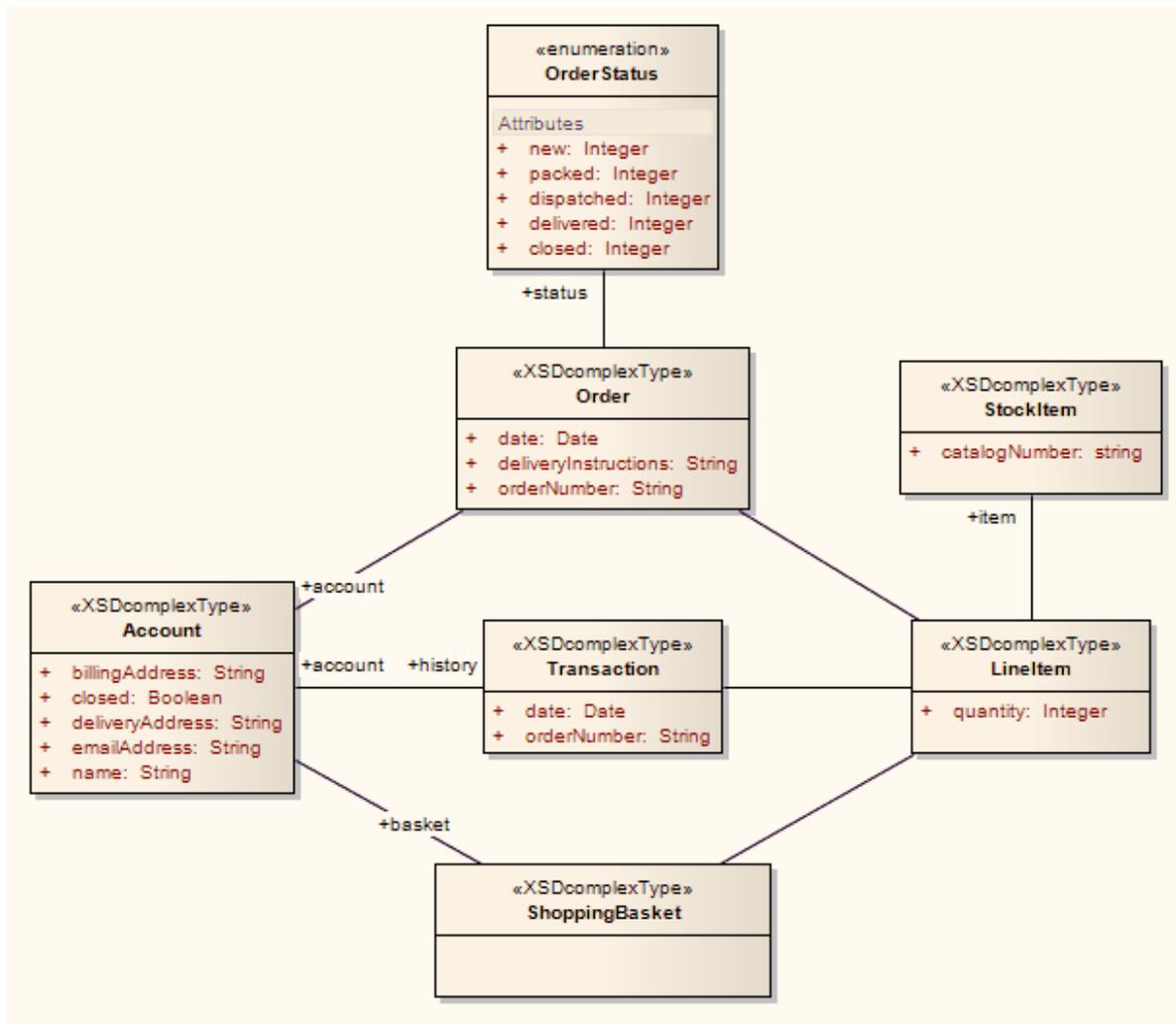
The purpose of the **XSD transformation** is to convert Platform-Independent Model (PIM) elements to UML Profile for XML elements as an intermediary step to creating an XML Schema. Each selected PIM Class element is converted to an «XSDcomplexType» element.

Example:

The Platform-Independent Model (PIM):



After transformation becomes the PSM:



Which in turn generates the following XSD:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="Account" type="Account" />
  <xs:complexType name="Account" >
    <xs:sequence>
      <xs:element name="name" type="xs:string" />
      <xs:element name="billingAddress" type="xs:string" />
      <xs:element name="emailAddress" type="xs:string" />
      <xs:element name="closed" type="xs:boolean" />
      <xs:element name="deliveryAddress" type="xs:string" />
      <xs:element ref="Order" />
      <xs:element ref="ShoppingBasket" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="LineItem" type="LineItem" />
  <xs:complexType name="LineItem" >
    <xs:sequence>
      <xs:element name="quantity" type="xs:integer" />
      <xs:element ref="StockItem" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Order" type="Order" />
  <xs:complexType name="Order" >
  
```

```

    <xs:sequence>
      <xs:element name="date" type="xs:date"/>
      <xs:element name="deliveryInstructions" type="xs:string"/
>
      <xs:element name="orderNumber" type="xs:string"/>
      <xs:element ref="LineItem"/>
      <xs:element name="status" type="OrderStatus"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="OrderStatus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="new"/>
      <xs:enumeration value="packed"/>
      <xs:enumeration value="dispatched"/>
      <xs:enumeration value="delivered"/>
      <xs:enumeration value="closed"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="ShoppingBasket" type="ShoppingBasket"/>
  <xs:complexType name="ShoppingBasket">
    <xs:sequence>
      <xs:element ref="LineItem"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="StockItem" type="StockItem"/>
  <xs:complexType name="StockItem">
    <xs:sequence>
      <xs:element name="catalogNumber" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Transaction" type="Transaction"/>
  <xs:complexType name="Transaction">
    <xs:sequence>
      <xs:element name="date" type="xs:date"/>
      <xs:element name="orderNumber" type="xs:string"/>
      <xs:element ref="Account"/>
      <xs:element ref="LineItem"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

Learn More:

- For more information, see the [Generate XSD](#) [1598] topic

9.3 Transformation Templates

Note that the Transformation Template mechanism is based very strongly on the Code Generation Template mechanism. For further information on Transformation Templates see the Code Template Editor section and also - for information on the Common Code Editor and intellisense - the Code Editors topic.

Access: [Settings | Transformation Templates](#)

To modify Transformation templates, follow the steps below

How To:

Step	Action	See also
1	Select the Settings Transformation Templates menu option. The Transformation Templates Editor displays.	
2	In the Language field, type or select the name of the transformation to modify.	
3	Select a template from the Templates list, and edit its contents in the editor pane.	
4	Click on the Save button.	

Reference:

Field	Usage	See also
Language	Select the name of the transformation.	
Template	Display the contents of the active template. Provide the editor for modifying templates.	
Templates	List the base transformation templates. The active template is highlighted. The Modified field indicates whether you have changed the default template for the current transformation.	
New Transformation Type	Create a new transformation.	
Stereotype Overrides	List the stereotyped templates, for the active base template. The Modified field indicates whether you have modified a default stereotyped template.	
Add New Stereotyped Override	Invoke a dialog for adding a stereotyped template, for the currently selected base template.	
Add New Custom Template	Invoke a dialog for creating a custom stereotyped template.	
Help	Launch the Enterprise Architect Help topic for this dialog.	
Get Default	Update the editor display with the default version of the active	

Field	Usage	See also
Template	template.	
Save	Overwrite the active templates with the contents of the editor.	
Delete	If you have overridden the active template, delete the override and replace it with the corresponding default transformation template.	

Learn More:

- [Code Template Editor](#)^[1485]
- [Code Editors](#)^[1394]

9.3.1 Import Transformations

Access: Project | Model Import/Export | Import Reference Data

Use To:

- Transfer **Transformation templates** between models

To import a Transformation template, follow the steps below

How To:

Step	Action	See also
1	Select the Project Model Import/Export Import Reference Data menu option. The Import Reference Data dialog displays.	
2	Click on the Select File button and browse to a .XML file containing the required Transformation template.	
3	Select the name of one or more template datasets and click on the Import button.	

9.4 Write Transformations

This topic provides help in writing your own transformations. Subjects covered are:

Topic	Detail	See also
Default Transformation Templates	Enterprise Architect provides a set of default transformation templates to customize to your requirements	Default Transformation Templates ^[1332]
General Syntax for the Intermediary Language	Describes the grammar for the transformation	General Syntax for the Intermediary Language ^[1333]
Syntax for Creating Objects	Describes the syntax followed in creating objects or elements	Syntax for Creating Objects ^[1334]
Syntax for Creating Connectors	Describes the syntax followed in creating connectors	Syntax for Creating Connectors ^[1335]
Transforming Duplicate Information	Describes how information is copied during transformation	Transforming Duplicate Information ^[1340]
Converting Types	Describes methods for converting data types to different target platform types	Converting Types ^[1340]
Converting Names	Describes methods for converting names of elements to different target platform naming conventions	Converting Names ^[1341]
Cross References	Provides cross verification of transformed elements	Cross References ^[1342]
Code template Framework	Provides forward engineering of UML models	Code Template Framework ^[1482]
Code Editors	Provides the list of editors supported by Enterprise Architect	Code Editors ^[1394]

Notes:

- Further hints and tips can be gleaned from a close study of the Transformation Templates provided with Enterprise Architect; writing transformations is very similar to writing code generation templates, so an understanding of the Code Template Framework can greatly assist in understanding transformations
- The Transformation Template editor provides the facilities of the Common Code Editor
- Transformation Templates are accessed from the **Settings | Transformation Templates** menu option

9.4.1 Default Transformation Templates

In most transformations, there is a lot of information that is simply copied to the target model. In order to make writing new transformations simpler, Enterprise Architect provides a default set of transformation templates. These templates perform a simple copy of the source model to the target model. This means that in order to write a new transformation you can modify the default templates to make the required changes.

Notes:

- When creating a new transformation you must modify at least one template before the transformation becomes available

9.4.2 Intermediary Language

All transformations in Enterprise Architect work by generating an intermediary text form of the model to generate.

Topics:

Topic	Detail	See also
Usage	<p>Any element is represented in this language by the type of element (for example, Class, Action, Method, Generalization or Tag) followed by the properties of the element and the elements that it is made from; the grammar for this resembles the following:</p> <pre> element : element Name { (element Property element) * } element Property: packageName stereotype propertyName = " propertyValueSymbol * " packageName: name = " propertyValueSymbol * " (. " propertyValueSymbol * ") * stereotype: stereotype = " propertyValueSymbol * " (, " propertyValueSymbol * ") * propertyValueSymbol : \ \ \ " Any character except " (U+0022), \ (U+005C) • <i>elementName</i> is any one of the set of element types • <i>propertyName</i> is any one of the set of properties Literal strings can be included in property values by escaping a quote character; for example: default = "\" Some string value.\""</pre>	<p>Objects <small>[1333]</small></p> <p>Connectors <small>[1338]</small></p>

9.4.3 Objects

Topics:

Topic	Detail	See also
Objects	<p>Objects are created in Enterprise Architect by generating text in the following form:</p> <pre> object Type { object Properties } where:</pre>	<p>XRef element <small>[1342]</small></p>

Topic	Detail	See also
	<p>obj ect Type is one of the following object types:</p> <ul style="list-style-type: none"> • <i>Action</i> • <i>ActionPin</i> • <i>Activity</i> • <i>ActivityParameter</i> • <i>ActivityPartition</i> • <i>ActivityRegion</i> • <i>Actor</i> • <i>Association</i> • <i>Change</i> • <i>Class</i> • <i>Collaboration</i> • <i>CollaborationUse</i> • <i>Component</i> • <i>DeploymentSpecification</i> • <i>DiagramFrame</i> • <i>Decision</i> • <i>EntryPoint</i> • <i>Event</i> • <i>ExceptionHandler</i> • <i>ExecutionEnvironment</i> • <i>ExitPoint</i> • <i>ExpansionNode</i> • <i>ExpansionRegion</i> • <i>ExposedInterface</i> • <i>GUIElement</i> • <i>InteractionFragment</i> • <i>InteractionOccurrence</i> • <i>InteractionState</i> • <i>Interface</i> • <i>InterruptibleActivityRegion</i> • <i>Issue</i> • <i>Iteration</i> • <i>Object</i> • <i>ObjectNode</i> • <i>MessageEndpoint</i> • <i>Node</i> • <i>Package</i> • <i>Parameter</i> • <i>Part</i> • <i>Port</i> • <i>ProvidedInterface</i> • <i>RequiredInterface</i> • <i>Requirement</i> • <i>Sequence</i> • <i>State</i> • <i>StateMachine</i> • <i>StateNode</i> • <i>Synchronization</i> • <i>Table</i> • <i>TimeLine</i> 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • <i>Trigger</i> • <i>UMLDiagram</i> • <i>UseCase</i> <p>objectProperties is zero, or one or more of the following properties:</p> <ul style="list-style-type: none"> • <i>Abstract</i> • <i>Alias</i> • <i>Arguments</i> • <i>Author</i> • <i>Cardinality</i> • <i>Classifier</i> • <i>Complexity</i> • <i>Concurrency</i> • <i>Filename</i> • <i>Header</i> • <i>Import</i> • <i>IsActive</i> • <i>IsLeaf</i> • <i>IsRoot</i> • <i>IsSpecification</i> • <i>Keyword</i> • <i>Language</i> • <i>Multiplicity</i> • <i>Name</i> • <i>Notes</i> • <i>Persistence</i> • <i>Phase</i> • <i>Scope</i> • <i>Status</i> • <i>Stereotype</i> • <i>Version</i> • <i>Visibility</i> <p>and zero, or one or more of the following elements:</p> <ul style="list-style-type: none"> • <i>Attribute</i> • <i>Classifier</i> • <i>Parameter</i> • <i>Operation</i> • <i>Parent</i> • <i>Tag</i> • <i>XRef</i> • <i>Any object</i> <ul style="list-style-type: none"> • Some of the above only apply to certain object types • Every object created in a transformation should include an XRef element, as it enables Enterprise Architect to synchronize with the element and makes it possible to create a connector to that Class in a transformation 	
Classes	<p>A simple Class can be created as follows:</p> <pre>Cl ass</pre>	

Topic	Detail	See also
	<pre> { name = " Example" } </pre> <p>It is then easy to add to this; the following example sets the language to C++, and adds a Tagged Value and an attribute:</p> <pre> Class { name = " Example" language = " C++" Tag { name = " default CollectionClass" value = " List" } Attribute { name = " count" type = " int" } } </pre>	
Attributes	<p>Attributes are created with the same structure as objects, and include the following properties:</p> <ul style="list-style-type: none"> • <i>Alias</i> • <i>Classifier</i> • <i>Collection</i> • <i>Container</i> • <i>Containment</i> • <i>Constant</i> • <i>Default</i> • <i>Derived</i> • <i>LowerBound</i> • <i>Name</i> • <i>Notes</i> • <i>Ordered</i> • <i>Scope</i> • <i>Static</i> • <i>Stereotype</i> • <i>Type</i> • <i>UpperBound</i> • <i>Volatile</i> <p>and the following elements:</p> <ul style="list-style-type: none"> • <i>Classifier</i> • <i>Tag</i> • <i>XRef</i> 	
Operations	<p>Operations are created with the same structure as objects, and include the following properties:</p> <ul style="list-style-type: none"> • <i>Abstract</i> • <i>Alias</i> • <i>Behavior</i> • <i>Classifier</i> • <i>Code</i> • <i>Constant</i> 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • <i>IsQuery</i> • <i>Name</i> • <i>Notes</i> • <i>Pure</i> • <i>ReturnArray</i> • <i>Scope</i> • <i>Static</i> • <i>Stereotype</i> • <i>Type</i> <p>and the following elements:</p> <ul style="list-style-type: none"> • <i>Classifier</i> • <i>Parameter</i> • <i>Tag</i> • <i>XRef</i> 	
Parameters	<p>Parameters are created with the same structure as objects, and include the Tag element and the following properties:</p> <ul style="list-style-type: none"> • <i>Classifier</i> • <i>Default</i> • <i>Fixed</i> • <i>Name</i> • <i>Notes</i> • <i>Kind</i> • <i>Stereotype</i> 	
Packages	<p>Packages differ from other objects in the following ways:</p> <ul style="list-style-type: none"> • A reduced set of properties of <i>alias</i>, <i>author</i>, <i>name</i>, <i>namespaceRoot</i>, <i>notes</i>, <i>scope</i>, <i>stereotype</i> and <i>version</i> • The extra property <i>namespaceRoot</i> • Must have a name specified • <i>Name</i> can be a qualified name; when a qualified name is specified the properties given are applied only to the final package • Can contain other packages • Can't contain attributes and operations 	
Tables	<p>Tables are a special sort of object, with the following differences from other object types:</p> <ul style="list-style-type: none"> • Can include columns and primary keys • Cannot include attributes 	
Columns	<p>Columns are similar to attributes, but have an <i>autonumber</i> element containing <i>Startnum</i> and increment, and the following added properties:</p> <ul style="list-style-type: none"> • <i>Length</i> • <i>NotNull</i> • <i>Precision</i> • <i>PrimaryKey</i> • <i>Scale</i> • <i>Unique</i> <p>In the column definition, you cannot assign a value to the <i>NotNull</i>, <i>PrimaryKey</i> or <i>Unique</i> properties</p>	

9.4.4 Connectors

Creating connectors in a transformation can be complex, but the process has the same form as creating elements. The difference is that you must also specify each end.

Topics:

Topic	Detail	See also
Connector Types	<p>The different connectors that can be created are as follows:</p> <ul style="list-style-type: none"> • Aggregation • Assembly • Association • Collaboration • ControlFlow • Connector • Delegate • Dependency • Deployment • ForeignKey • Generalization • InformationFlow • Instantiation • Interface • InterruptFlow • Manifest • Nesting • NoteLink • ObjectFlow • Package • Realization • Sequence • Transition • UseCase • Uses <p>There are two different types of Class that you can use as a connector end: one created by a transformation, and one for which you already know the GUID</p>	
Connect to a Class Created by a Transformation	<p>The most common connection is to connect to a Class created by a transformation; to do this you must have three items of information:</p> <ul style="list-style-type: none"> • The original Class GUID • The name of the transformation • The name of the transformed Class <p>This type of connector is created using the TRANSFORM_REFERENCE function macro; when the element is in the current transformation, it can be safely omitted from the transformation</p> <p>The simplest example of this is when you have created multiple Classes from a single Class in a transformation and want a connector between them; consider this example from the EJB Entity transformation:</p> <pre> Dependency { </pre>	Cross References <small>[1342]</small>

Topic	Detail	See also
	<pre> %TRANSFORM_REFERENCE(" EJBReal i zeHome" , classGUI D) % st ereot ype=" EJBReal i zeHome" Source { %TRANSFORM_REFERENCE(" EJBEnt i tyBean" , classGUI D) % } Target { %TRANSFORM_REFERENCE(" EJBHomeI nt erface" , classGUI D) % } } </pre> <p>There are three uses of the TRANSFORM_REFERENCE macro: one to identify this connector for synchronization purposes and the other two to identify the ends; all three use the same source GUID, because they all come from the one original Class</p> <p>None of the three have to specify the transformation because the two references are referencing something in the current transformation - each of them then only has to identify the transform name</p> <p>Of course, it is also possible to create a connector from another connector</p> <p>You can create a connector template and list over all connectors connected to a Class from the Class level templates; you don't have to worry about only generating it once, because if you have created a TRANSFORM_REFERENCE for the connector then Enterprise Architect automatically synchronizes them</p> <p>The following copies the source connector:</p> <pre> %connector Type% { %TRANSFORM_CURRENT() % %TRANSFORM_REFERENCE(" Connector" , connector GUI D) % Source { %TRANSFORM_REFERENCE(" Cl ass" , connector Sour ceGUI D) % %TRANSFORM_CURRENT(" Sour ce") % } Target { %TRANSFORM_REFERENCE(" Cl ass" , connector Dest GUI D) % %TRANSFORM_CURRENT(" Tar get ") % } } </pre>	
<p>Connecting to a Class for which you know the GUID</p>	<p>The second type of Class that you can use as a connector end is one for which you know the current GUID</p> <p>To do this, specify the GUID of the target Class in either the source or target end; the following example creates a dependency from a Class created in a transformation, to the Class it was transformed from:</p> <pre> Dependency { %TRANSFORM_REFERENCE(" Sour ceDependency" , classGUI D) % st ereot ype=" t ransf or medFr om" Source { </pre>	

Topic	Detail	See also
	<pre> %TRANSFORM_REFERENCE(" Cl ass" , cl ass GUI D) % } Target { GUI D=%qt %%c l ass GUI D%%qt % } </pre>	

Notes:

- *ForeignKey* is a special case where not just a connector is created - you must also list the columns involved in the transformation; in addition, tags specified on the connector are actually created on the foreign key operation in the source Class, and a cascade property can be added - for example, *cascade="update","delete"*
- Also, the *nullable* property of the foreign key is set to **true** if either end has a multiplicity that includes zero; this can be overridden by setting *nullable="1"* or *nullable="0"* in the source section of the foreign key
- Each connector is transformed at both end objects, therefore the connector might appear twice in the transformation; this is nothing to be concerned about, but you should check carefully that the connector is generated exactly the same way, regardless of which end is on the current Class

9.4.5 Copy Information

In many transformations there is a substantial amount of information to be copied.

It would be tedious to type all of the common information into a template so that it is copied to the transformed Class; the alternative is to use the **TRANSFORM_CURRENT** function macro.

Reference:

Topic	Detail	See also
TRANSFORM_CURRENT	<p>TRANSFORM_CURRENT(<listOfExcludedItems>)</p> <p>The function generates an exact copy of all the properties of the current item, except for the items named in <i><listOfExcludedItems></i></p> <p>Another form of the function is available when transforming connectors that enables either end of the connector to be copied:</p> <p>TRANSFORM_CURRENT(<connectorEnd>,<listOfExcludedItems>)</p> <p>This generates an exact copy of the connector end specified by <i><connectorEnd></i> except for the items named in <i><listOfExcludedItems></i>, where <i><connectorEnd></i> is either Source or Target</p>	

9.4.6 Convert Types

Topics:

Topic	Detail	See also
Synopsis	Different target platforms almost certainly require different data types, so you often require a method of converting between types, as offered by the following macro:	

Topic	Detail	See also
	<ul style="list-style-type: none"> • CONVERT_TYPE(<i><destinationLanguage></i>, <i><originalType></i>) <p>This function converts <i><originalType></i> to the corresponding type in <i><destinationLanguage></i>, using the datatypes and common types defined in the model, where <i><originalType></i> is assumed to be a platform independent common type</p> <p>A similar macro is available when transforming common datatypes to the datatypes for a specified database:</p> <ul style="list-style-type: none"> • CONVERT_DB_TYPE(<i><destinationDatabase></i>, <i><originalType></i>) <p>This function converts <i><originalType></i> to the corresponding datatypes in <i><destinationDatabase></i>, which is defined in the model; <i><originalType></i> refers to a platform independent common datatype</p>	

9.4.7 Convert Names

Different target platforms use different naming conventions. As a result you might not want to copy the names of your elements directly into the transformed models. To facilitate this requirement, Enterprise Architect's transformation templates provide a CONVERT_NAME function macro.

Another way in which you can transform a name is to remove a prefix from the original name, with the REMOVE_PREFIX macro.

Topics:

Topic	Detail	See also
CONVERT_NAME <i><originalName></i> , <i><originalFormat></i> , <i><targetFormat></i>)	<p>This macro converts <i><originalName></i>, which is assumed to be in <i><originalFormat></i>, to <i><targetFormat></i>.</p> <p>The supported formats are:</p> <ul style="list-style-type: none"> • Camel Case: New words start with a capital letter except for the first word, which begins with a lower case letter; for example, <i>myVariableTable</i> • Pascal Case: Same as Camel Case but the first letter of the first word is upper case; for example, <i>MyVariableTable</i> • Spaced: Words are separated by spaces; the case of letters is ignored • Underscored: Words are separated by underscores; the case of letters is ignored <p>The original format might also specify a list of delimiters to be used. For example a value of ' _ ' breaks words whenever either a space or underscore is found.</p> <p>The target format might also use a format string that specifies the case for each word and a delimiter between them. It takes the following form:</p> <p><i><firstWord></i>(<i><delimiter></i>)<i><otherWords></i></p> <ul style="list-style-type: none"> • <i><firstWord></i> controls the case of the first word (see below) • <i><delimiter></i> is the string generated between words • <i><otherWords></i> applies to all words after the first 	

Topic	Detail	See also
	<p>word</p> <p><<i>firstWord</i>> and <<i>otherWords</i>> are both a sequence of two characters. The first character represents the case of the first letter of that word, and the second character represents the case of all subsequent letters. An upper case letter forces the output to upper case, a lower case letter forces the output to lower case, and any other character preserves the original case.</p> <p>Example 1: To capitalize the first letter of each word and separate multiple words with a space:</p> <p>" Ht () Ht " to output " My Var i abl e Tabl e "</p> <p>Example 2: To generate the equivalent of Camel Case, but reverse the roles of upper and lower case; that is, all characters are upper case except for the first character of each word <i>after</i> the first word:</p> <p>" HT() hT " to output " MY vARI ABLE t ABLE "</p>	
REMOVE_PREFIX (< <i>originalName</i> >, < <i>prefixes</i> >)	<p>This macro removes any prefix found in <<i>prefixes</i>> from <<i>originalName</i>>. The prefixes are specified in a semi-colon separated list.</p> <p>The macro is often used in conjunction with the <i>CONVERT_NAME</i> macro. For example, this code creates a get <i>property name</i> according to the options for Java.</p> <pre>\$pr oper t yName=%REMOVE_PREFIX(at t Name , genOpt Pr oper t yPr ef i x) % %i f genOpt GenCapi t al i sedPr oper t i es==" T" % \$pr oper t yName=%CONVERT_NAME (\$pr oper t yName, " camel case", " pascal case") % %endI f %</pre>	

Notes:

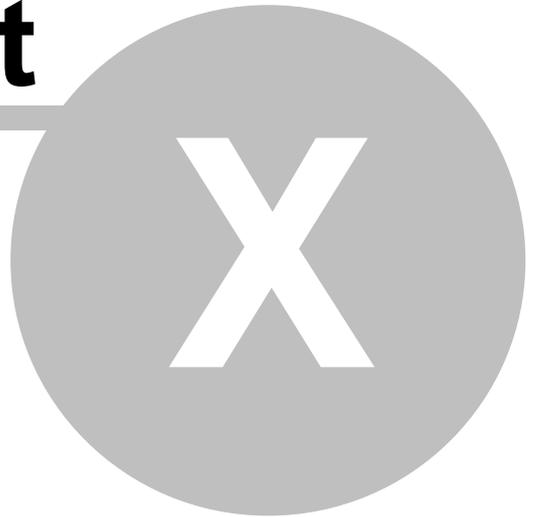
- Acronyms are not supported when converting from Camel Case or Pascal Case

9.4.8 Cross References**Topics:**

Topic	Detail	See also
Abstract	<p>Cross References are an important part of transformations. They are used to:</p> <ul style="list-style-type: none"> • Find the transformed Class to synchronize with • Create connectors between transformed Classes • Specify a classifier of a type • Determine where to transform to for future transformations <p>Each cross reference has three different parts:</p> <ul style="list-style-type: none"> • A <i>Namespace</i>, corresponding to the transformation that generated the element 	<p>DDL Transformation [1310]</p>

Topic	Detail	See also
	<ul style="list-style-type: none"> • A <i>Name</i>, which is a unique reference to something that can be generated in the above transformation • A <i>Source</i>, which is the GUID of the element that this element was created from <p>When writing the templates for a transformation, it is easiest to create the cross references using the TRANSFORM_REFERENCE macro that is defined for this purpose. It has three optional parameters.</p> <p>TRANSFORM_REFERENCE(<name>, <sourceGuid>, <namespace>)</p> <p>Generates a reference that can be used in the ways described above. It resembles the following.</p> <pre>XRef{namespace="<namespace>" name="<name>" source="<sourceGuid>"</pre> <p>Where:</p> <ul style="list-style-type: none"> • If <name> is not specified it gets the name of the current template • If <sourceGUID> is not specified it gets the GUID of the current Class • If <namespace> is not specified it gets the name of the current transformation <p>The only time that this should be specified is when creating a connector to a Class created in a different transformation.</p> <p>A good example of the use of cross references is in the DDL templates provided with Enterprise Architect. In the Class template a cross reference is created with the name table. Then up to two different connectors can be created, each of which must identify the two Classes it connects using cross references while having its own unique cross reference.</p>	
Specify Classifiers	<p>Objects, attributes, operations and parameters can all reference another element in the model as their type. When this type is created from a transformation you must use a cross reference to specify it, using the TRANSFORM_CLASSIFIER macro.</p> <p>TRANSFORM_CLASSIFIER(<name>, <sourceGuid>, <namespace>)</p> <p>Generates a cross reference within a classifier element, where the parameters are identical to the TRANSFORM_REFERENCE macro but the name Classifier is generated instead of <i>XRef</i>.</p> <p>If the target classifier already exists in the model before the transformation, a TRANSFORM_CLASSIFIER is inappropriate and instead the GUID can be given directly to a classifier attribute.</p> <p>If a classifier is specified for any type it overrides the type specified.</p>	

Part



10 Database Engineering



This section describes database engineering, explaining:

- How to import database schema, and
- How to generate DDL for the model

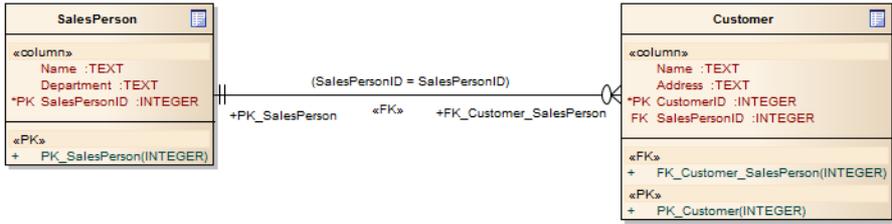
Learn More:

- [Import database schema](#)^[1375]
- [Generate DDL](#)^[1379]
- [Physical Data Modeling](#)^[1346]

10.1 Physical Data Model

The Physical data model visually represents the structure of the data as implemented by a relational database schema.

Topics:

Topic	Detail	See also
Benefit	In addition to providing a visual abstraction of the database structure, an important benefit of defining a Physical data model is that you can automatically derive the database schema from the model; this is possible due to the richness of meta-data captured by a Physical data model and its close mapping to aspects of the database schema, such as tables, columns, primary and foreign keys	
Example	<p>The following is an example of a Physical data model that could be used to automatically generate a database schema</p> <p>Each table is represented by a UML Class; columns, including Primary and Foreign Keys, are modeled using UML Attributes and Operations</p>  <p>A Physical data model defined using a UML Profile and the Information Engineering notation</p> <p>The above model is defined using Enterprise Architect's <i>UML Profile for Data Modeling</i>; the relationship between the tables uses the Information Engineering notation</p> <p>Information Engineering is one of three notations that Enterprise Architect supports to help Data Modelers distinguish cardinality in relationships</p>	

Learn more:

- [Tables and Columns](#) ^[1347]
- [Data Types](#) ^[1353]
- [Primary and Foreign Keys](#) ^[1357]
- [Stored Procedures](#) ^[1366]
- [Indexes](#) ^[1366]
- [Triggers](#) ^[1369]
- [Check Constraints](#) ^[1370]
- [Views](#) ^[1370]
- Supported Databases
- [Oracle Packages](#) ^[1373]
- [Forward and Reverse Engineer Databases](#) ^[1345]
- [Transform Abstract Models to Physical Data Models Using MDA](#) ^[1310]

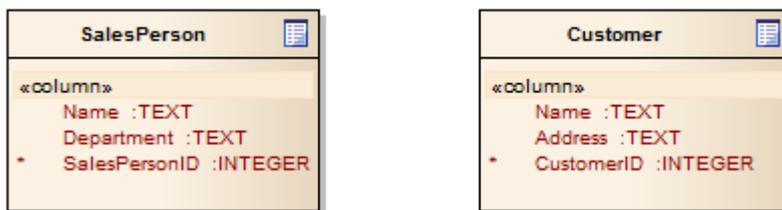
10.1.1 Tables and Columns

The basic structural element of a relational database is the *table*, which represents a set of records, or rows, that have the same structure. Each row contains one or more *columns*. Every individual item of data entered into a relational database is represented by a value in a column of a row in a table. Enterprise Architect's *UML Profile for Data Modeling* represents database tables as Classes with a stereotype of <<table>>. Columns are represented by Attributes with a stereotype of <<column>>. Tables and columns form the basis of Physical data models.

Enterprise Architect can generate simple DDL scripts to create the tables in your model. You can also perform Model Driven Architecture (MDA) Transformations to DDL - Enterprise Architect provides a template specifically for DDL transformations.

To help you map Class Attributes to Table fields, you can create connectors between specific attributes (features) in the Class element and the column Attributes in the Table element.

A simple example of a Physical data model in Enterprise Architect is shown below. The example consists of two tables represented by UML Classes, named *SalesPerson* and *Customer*. The table stereotype is denoted by the icon in the top-right corner of each Class. The tables each define three columns using UML Attributes typed appropriately for the target DBMS, which happens to be MySQL.



A simple Data model consisting of two tables, represented by UML Classes

So far, Primary Key and Foreign Key information modeling is not discussed; this is considered in later topics.

Learn more:

- [Create a Table](#)^[1347]
- [Model Transformation](#)^[1301]
- [DDL Transformation](#)^[1310]
- [Connect to Element Feature](#)^[742]
- [Create Columns](#)^[1351]
- [Primary and Foreign Keys](#)^[1357]

10.1.1.1 Create a Table

How To:

To add a Table to your data model, follow the steps below:

Step	Action	See Also
1	Open a diagram.	
2	Select the More tools Data Modeling menu option on the Toolbox.	

Step	Action	See Also
3	Click on the <i>Table</i> element in the list of elements, then click on the diagram. The Table element is displayed on the diagram.	
4	Use the Table Properties dialog to set the table name and other properties as required.	

Learn more:

- [Set the properties of a Table](#) ^[1348]

10.1.1.1.1 Working with Table Properties

Once you have created a table, you must set its properties to define database-specific information that can be used to generate the schema in DDL.

Most table properties, including the database type, are set using the Properties dialog. Some additional properties are defined using Tagged Values, such as the Table Owner and DBMS-specific options for MySQL and Oracle.

Once the table properties are defined, you are ready to add columns.

Topics:

Topic	Link
<p>Set the Database type for a table - other than the table name, the most important property to set for a table is the database type, which defines:</p> <ul style="list-style-type: none"> • The list of data types that are available for defining columns, and • Which dialect of DDL is generated 	Set the Database Type ^[1348]
<p>Set the Table Owner - Some relational databases enable you to assign a table to one of several <i>owners</i> within the database; Enterprise Architect enables you to model this using the <i>Owner</i> Tagged Value, so that in the generated DDL script the table name is prefixed by the owner name</p>	Set Table Owner/Schema ^[1349]
<p>Example MySQL options - To make use of foreign keys in MySQL, you must declare the table type as InnoDB</p>	Set MySQL Options ^[1349]
<p>Example Oracle options - To set additional Oracle table properties, use the table's Tagged Values</p>	Set Oracle Table Properties ^[1350]

Learn more:

- [Create Columns](#) ^[1351]

10.1.1.1.1.1 Set the Database Type

The most important property to set for a table (after its name) is the database type. This defines the list of data types that are available for defining columns, and also declares which dialect of DDL is generated.

Access: **Properties Dialog (Table element) > General page: Database**

How To:

Step	Action	See Also
1	Open the Property dialog for a Table element	
2	Locate the Database field on the General page	
3	Drop-down the list and select the required database type	
4	Save changes by clicking on the Apply or OK buttons	

10.1.1.1.1.2 Set Table Owner/Schema

Some relational databases enable you to assign a table to one of several schemas (or *owners*) within the same database. Enterprise Architect enables you to model this information using the *Owner* Tagged Value. By setting this optional value, you can tailor the generated DDL script, causing the table name to be prefixed by it's assigned owner name.

How To:

Step	Action	See Also
1	Add a Tagged Value named <i>Owner</i> , to the Table element	
2	Type the owner name for this table in the Value field of the Tagged Value Changes are automatically saved	

Notes:

- For a PostgreSQL database, you should instead create a Tagged Value named OWNER TO with the corresponding value

Learn More:

- [How to Add a Tagged Value](#)^[764]

10.1.1.1.1.3 Set MySQL Options

To make use of foreign keys in MySQL, you must declare the table type as InnoDB.

How To:

Step	Action	See Also
1	Add a Tagged Value named Type , to the table	
2	Set the Value field to InnoDB	

When you generate DDL for this table, the table type is included in the SQL script.

To allow for later versions of MySQL, additional table options that can be added in the same manner include:

Tag	Value (Example)
ENGINE	InnoDB
CHARACTER SET	latin1
CHARSET	latin1
COLLATE	latin1_german2_ci

Learn More

- [How to Add a Tagged Value](#)^[764]

10.1.1.1.4 Set Oracle Table Properties

To set additional Oracle table properties, you use the table's Tagged Values

How To:

Step	Action	See Also
1	Add one or more Tagged Values to the table, using the names provided in the Property/Tag column of the reference table below	
2	Specify the appropriate value for each tag Examples are provided in the Value column of the reference table below	

The same properties can be added to indexes and constraints, by highlighting the index or constraint Operation and adding the appropriate Tagged Values.

Reference:

Property/Tag	Value
BUFFER_POOL	DEFAULT
CACHE	NOCACHE
DBVERSION	9.0.111
FREELISTS	1
GRANT OWNER1	SELECT
GRANT OWNER2	DELETE, INSERT, SELECT, UPDATE
INITIAL	65536
INITRANS	1
LOGGING	LOGGING
MAXEXTENTS	2147483645
MAXTRANS	255

Property/Tag	Value
MINEXTENTS	1
MONITORING	MONITORING
OWNER	OWNER1
PARALLEL	NOPARALLEL
PCTFREE	10
PCTINCREASE	0
PCTUSED	0
SYNONYMS	PUBLIC:TABLE_PUB;OWNER2:TABLE_OWNER2
TABLESPACE	MY_TABLESPACE
TEMPORARY	YES

Learn More

- [How to Add a Tagged Value](#)^[764]

10.1.1.2 Create Columns

Columns are represented in the UML Data Modeling Profile as a stereotyped attribute; that is, an attribute with the *Column* stereotype.

How To:

To create columns, follow the steps below:

Step	Action	See Also
1	Right-click on the required Table in a diagram, and select the Attributes context menu option The Attributes dialog displays, showing the columns of the selected Table	
2	In the Name field, type the column name	
3	In the Data Type field, click on the drop-down arrow and select the data type (ensure the Table's target database is set first; this populates the Data Type drop-down list) Click on the Save button	Set the Database Type ^[1348] Oracle Data Types ^[1354]
4	Complete the required optional fields, listed in the reference table below	
5	Click on the Column Properties button The Database Columns Properties dialog displays	
6	If you require a sequence, such as an Oracle sequence, select the AutoNum property, set the value to True and, if necessary, define the start number and increment Click on the OK button to return to the Attributes dialog	

Step	Action	See Also
7	Click on the Save button	
8	Click on the: <ul style="list-style-type: none"> • New button to define another column or • Close button to exit from the dialog 	

Reference:

Field	Description	See also
Primary Key	Indicates whether the column is the primary key for this table	
Not Null	Indicates whether empty values are forbidden for this column	
Unique	Indicates whether it is forbidden for any two values of this column to be identical The unique <i>stereotype</i> applied to an index ensures that no two <i>combinations</i> of values across a <i>set</i> of columns can be identical	
Initial	A value that can be used as a default value for this column, if required	
Access	Determines the column's scope (Private , Protected or Public) The default value is Public	
Alias	An alternative name for the field for display and documentation purposes	
Notes	Additional information necessary to document the column You can format the text using the Notes toolbar at the top of the field	

Notes:

- For MySQL, before creating columns you must first add ENUM and SET datatypes

Learn more:

- [Reorder Columns](#) ^[1352]
- [Data Types](#) ^[1353]
- [MySQL Data Types](#) ^[1354]
- [Options - Code Editors](#) ^[1519]

10.1.1.2.1 Reorder Columns

You can change the order in which columns are listed in a table, to define the order in which columns appear in the generated DDL that defines your database schema.

How To:

To change the column order, follow the steps below:

Step	Action	See Also
1	In the Columns dialog, highlight a column name in the Columns panel	
2	Click on the: <ul style="list-style-type: none">  button to move the column up one position  button to move the column down one position 	

10.1.2 Data Types

Each column that you define in your data model has a data type that is appropriate to the information being stored by that column. Data types vary from one DBMS product to another and also vary between versions of a given database product. Therefore Enterprise Architect provides you with tools to add new data types to your model repository and to convert data types between DBMS products.

Learn more:

- [Add New Datatype to Your Model Repository](#)^[1353]
- [Map Data Types between DBMS products](#)^[1355]

10.1.2.1 Add New Datatypes

Using Enterprise Architect's Database Datatypes screen, you can add to the set of data types that are available for a particular DBMS. This enables you to include any new datatypes that are supported by later versions of the DBMS and not yet included with Enterprise Architect.

You can also add an entire DBMS product and its built-in data types. This is helpful if you want to create a physical data model for a DBMS product that is not yet supported natively by Enterprise Architect.

Access: [Settings | Database Datatypes](#)

How To:

To add a new datatype to a DBMS product, follow the steps below:

Step	Action	See Also
1	Select the Database Datatypes menu option The Database Datatypes screen displays	
2	In the Product Name field click on the drop-down arrow and select an existing DBMS If your DBMS is not listed, you can click the Add Product button to add it	
3	In the Datatype field, type a name for the data type	
4	Select the appropriate Size radio button and, if appropriate, specify the default and maximum values	
5	In the Common Type field, click on the drop-down arrow and select a common type to which your data type applies	
6	Click on the Save button to create your data type	

You can transport these data types between models using Enterprise Architect's Reference Data

Learn more:

- [Add MySQL Data Types](#) ^[1354]
- [Customize Oracle Data Types](#) ^[1354]
- [Export Data Types as Reference Data](#) ^[237]
- [Import Data Types as Reference Data](#) ^[239]

10.1.2.1.1 MySQL Data Types

MySQL supports the **ENUM** and **SET** data types, which must be added to your Enterprise Architect model before you can use them as the types for columns.

Access: [Settings | Database Datatypes](#)

How To:

To add the ENUM and SET data types for MySQL, follow the steps below:

Step	Action	See Also
1	Select the Database Datatypes menu option The Database Datatypes dialog displays	
2	In the Product Name field select MySQL	
3	Add the data types ENUM and SET	

When using these data types later in a Column's **Initial** field, type the values as a comma-separated list, in the format:

('one','two','three')

If one value is the default, use the format:

('one','two','three') default 'three'

10.1.2.1.2 Oracle Data Types

The Oracle data types **NUMBER** and **VARCHAR** have additional properties that you can model.

Reference:

Data Type	Detail	See also
NUMBER	The NUMBER data type requires precision and scale properties The Precision and Scale fields are displayed on the Attributes dialog when the data type is set to NUMBER ; if you enter information into these fields, it is displayed on your diagrams For example: create NUMBER by setting Precision = 0 and Scale = 0	

Data Type	Detail	See also
	create NUMBER(8) by setting Precision = 8 and Scale = 0 create NUMBER(8,2) by setting Precision = 8 and Scale = 2	
VARCHAR	Oracle VARCHAR2(15 CHAR) and VARCHAR2(50 BYTE) data types can be created by adding the Tagged Value LengthType with the value CHAR or BYTE	

Learn more:

- [How to Add a Tagged Value](#)^[764]

10.1.2.2 Map Data Types between DBMS products

One of the advantages of using visual models to hide implementation details is that you can more easily change the target technology or platform when required. Enterprise Architect provides tools such as Datatype Mapping to help you do this for physical data models. For example, your project might involve migration from one DBMS platform to another.

After reverse engineering your database into a physical data model, you must remap the data types before generating the schema for the new DBMS product. Enterprise Architect provides a set of default mappings to help you automate the conversion process. However, you might want to customize the default mappings to suit your specific project requirements.

Access: [Settings | Database Datatypes: Datatype Map](#)

Use to:

- Streamline conversion of one implementation-specific database product to another
- Map data types from one DBMS product to another
- Maximize portability of your database designs
- Customize default mappings

How To:

To map data type sizes between databases, follow the steps below:

Step	Action	See Also
1	In the From Product Name field, click on the drop-down arrow and select the DBMS product to map data types from The Defined Datatypes for Databases panel displays all the defined data types for the product and, where appropriate, their sizes and values	
2	Click on the data type to map - this must have a defined size unit and value The Datatype and Common Type fields under the From Product Name field display this data type	
3	In the To Product Name field, click on the drop-down arrow and select the DBMS product to map data types to The Datatype and Common Type fields under this field display the corresponding values to those in the fields for the 'From' product	
4	In the Size panel, click on the radio button for the appropriate size unit and	

Step	Action	See Also
	type the default values in the corresponding data fields	
5	Click on the Save button to save the mapping.	

You can repeat this process for all the data types you want to map.

Once you are satisfied with the data type mappings, you can then convert either individual tables or an entire package of tables to the new target DBMS product.

10.1.2.2.1 Data Type Conversion Procedure

Once a database schema has been set up on an Enterprise Architect diagram (either by importing through ODBC or manually setting up the tables), the DBMS can be changed to another type and the column datatypes are mapped accordingly.

To map the DBMS type of a table to another DBMS type

How To:

Step	Action	See Also
1	Double-click on the Table element in a diagram The Table Properties dialog displays	
2	The Database field shows the current DBMS for this table.	
3	To map the column datatypes to another DBMS, select the target from the Database drop-down and click on the Apply button.	
4	The datatypes are converted to match those of the new DBMS, and these are reflected in any DDL generated from this table.	

10.1.2.2.2 Data Type Conversion for a Package

The DBMS Package procedure or mapper enables you to convert a package of database tables from one DBMS type to another DBMS type, as well as providing the ability to change the ownership of tables.

Access: [Project Browser Package Context Menu | Code Engineering | Reset DBMS Options > Manage DBMS Options](#)

How To:

To map the DBMS types of a package to another DBMS type, follow the steps below:

Step	Action	See Also
1	In the Current DBMS field, click on the drop-down arrow and select the current DBMS	
2	In the New DBMS field click on the drop-down arrow and select the target DBMS	
3	Select the Convert DBMS Type checkbox	

Step	Action	See Also
4	If there are child packages that also require changing, select the Process Child Packages checkbox	
5	Click on the OK button All tables in the selected packages are mapped to the new DBMS	

To change the owner of the table or all of the tables in a package, follow the steps below:

Step	Action	See Also
1	In the New Owner field, type the name of the new table owner	
2	In the Current Owner field, click on the drop-down arrow and select either: <ul style="list-style-type: none"> The current owner to change, or <All> to change the ownership of all tables in the package to the name you typed in the New Owner field 	
3	Select the Change Table Owner checkbox.	
4	If there are child packages that also require changing, select the Process Child Packages checkbox	
5	Click on the OK button The ownership changes for all Tables in the selected packages that have the specified current owner	

Learn More:

- [Set Table Owner](#) ^[1349]
- [Diagram Properties](#) ^[569]

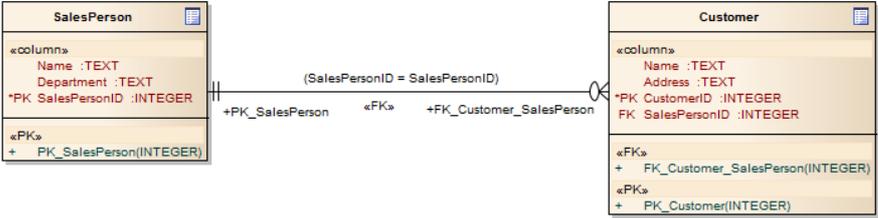
10.1.3 Database Keys

Two types of key are used to access tables in a relational database: *Primary Keys* and *Foreign Keys*.

- A Primary Key uniquely identifies a record in a table
- A Foreign Key accesses data in some other related table via its Primary Key

Topics:

Topic	Detail	See also
General Usage	In Enterprise Architect, Primary Keys and Foreign Keys are represented by stereotyped UML attributes and operations The following example shows a data model that includes both Primary Key and Foreign Key columns	

Topic	Detail	See also
Example	 <p>A Physical data model defined using a UML Profile that includes Primary and Foreign Keys</p>	
Primary Key	In each table, one of the columns is designated as the Primary Key, indicated by the <i>PK</i> label beside the column, and a corresponding Primary Key Operation	
Foreign Keys	Foreign Keys are shown denoted by the <i>FK</i> labels	
Association Notation	<p>The Association that defines a Foreign Key relationship between <i>SalesPerson</i> and <i>Customer</i> uses the Information Engineering notation; this is one of three notations that Enterprise Architect supports to help Data Modelers distinguish cardinality in relationships</p> <p>Enterprise Architect also supports:</p> <ul style="list-style-type: none"> • IDEF1X notation • UML notation 	

Learn more:

- [Primary Keys](#) ^[1358]
- [Foreign Keys](#) ^[1361]

10.1.3.1 Primary Key

A Primary Key uniquely identifies a record in a table and can consist of one or more columns.

Topics:

Topic	Detail	See also
Types of Primary Key	<p>In Enterprise Architect, a:</p> <ul style="list-style-type: none"> • Simple Primary Key (single column) is defined as the attribute of a stereotyped operation • Complex Primary Key (several columns) is defined as the stereotyped operation itself <p>For example, a column <i>LastName</i> might not be unique within a table, so a Primary Key is created from the <i>LastName</i>, <i>FirstName</i> and <i>DateOfBirth</i> columns</p>	

Learn more:

- [Create a Primary Key](#) ^[1359]
- [Define a Primary Key Name Template](#) ^[1359]
- [Foreign Keys](#) ^[1361]

10.1.3.1.1 Create a Primary Key

How To:

To define a simple Primary Key consisting of a single column, follow the steps below:

Step	Action	See Also
1	Right-click on the table in a diagram and select the Attributes context menu option The Attributes dialog displays	
2	Select the column that makes up the Primary Key	
3	Select the Primary Key checkbox and click on the Save button A stereotyped operation is automatically created; it is this operation that defines the Primary Key for the table To remove a Primary Key, simply delete this operation	

To define a complex Primary Key, consisting of more than one column, follow the steps below:

Step	Action	See Also
1	Follow the steps above to create a Simple Primary Key; it doesn't matter which column you choose	
2	Right-click on the table in a diagram and select the Operations context menu option The Operations dialog displays	
3	Select the Primary Key operation (its name begins with PK_) and then click on the Column tab	
4	Click on the New button to add a column to the Primary Key, select a column from the Column Name list box, and then click on the Save button	
5	Click on the Hand buttons (up and down arrow) to change the order of columns in the Primary Key, if necessary	

Learn more

- [Creating Non-Clustered Primary Keys for SQL Server](#)^[1360]

10.1.3.1.2 Define Primary Key Name Template

Enterprise Architect enables you to customize the way Primary Keys are named by default. This is helpful if you have specific modeling conventions that stipulate how Primary Keys are documented.

By changing the **Primary Key Name Template**, you adjust the name that Enterprise Architect assigns to the UML Operation that represents the Primary Key.

Access: **Tools | Options | Source Code Engineering | Code Editors: DDL Name Templates**

How To:

To define the name template for a Primary Key, follow the steps below:

Step	Action	See Also
1	Click on the DDL Name Templates button The DDL Name Template dialog displays, showing the default name templates	
2	Edit or replace the template in the Primary Key Name Template field For example, if you want to display the Primary Key description as <i>PK_tablename_columnname</i> then change the Primary Key Name Template field to <i>PK_%tablename%_%columnname%</i>	
3		
4	Click on the Save button	

10.1.3.1.3 SQL Server Non Clustered Keys

When you create a primary key in SQL Server, it is created with a 'clustered index' by default. Therefore when you model a primary key for SQL Server in an Enterprise Architect data model, the corresponding DDL creates a clustered index for that primary key by default.

Clustered indexes provide improved performance for accessing the column(s) involved, but only one clustered index is allowed per table.

In some situations, you might be more interested in the performance of columns other than the Primary Key, and therefore must change the default assignment of the clustered index. Enterprise Architect allows you to model this.

How To:

To define a Primary Key as non-clustered for a SQL Server table, follow the steps below:

Step	Action	See Also
1	Right-click on the table in a diagram and select the Operations context menu option The Table Operations dialog displays	
2	Highlight the Primary Key Operation and click on the Extended Properties button The Database Operation Properties dialog displays	
3	Select the SQL Server Non Clustered Primary Key checkbox	
4	Click on the Save & Close button	

Subsequently, you can model an index for that table and define it as clustered instead.

Learn more

- [Indexes](#)^[1366]

10.1.3.2 Foreign Key

A Foreign Key accesses data in some other related table via its Primary Key.

Topics:

Topic	Detail	See also
The nature of Foreign Keys	A Foreign Key consists of a collection of columns (UML Attributes) that together have some operational meaning (they enforce a relationship to the Primary Key in the other table)	
Modeling	Foreign keys are modeled in Enterprise Architect as operations with the stereotype FK The operation's parameters become the columns involved in the key	
Necessity of Use	It is not necessary to define a Foreign Key just to access another table through its Primary Key Foreign Keys are a feature of some database management systems, providing 'extras' such as referential integrity checking that prevents the deletion of a record if its Primary Key value exists in some other table's Foreign Key The same thing can be achieved programmatically	

Learn more

- [Create a Foreign Key](#)^[1367]
- [Composite Foreign Key](#)^[1364]
- [Define a Foreign Key Name Template](#)^[1365]

10.1.3.2.1 Create a Foreign KeyHow To:

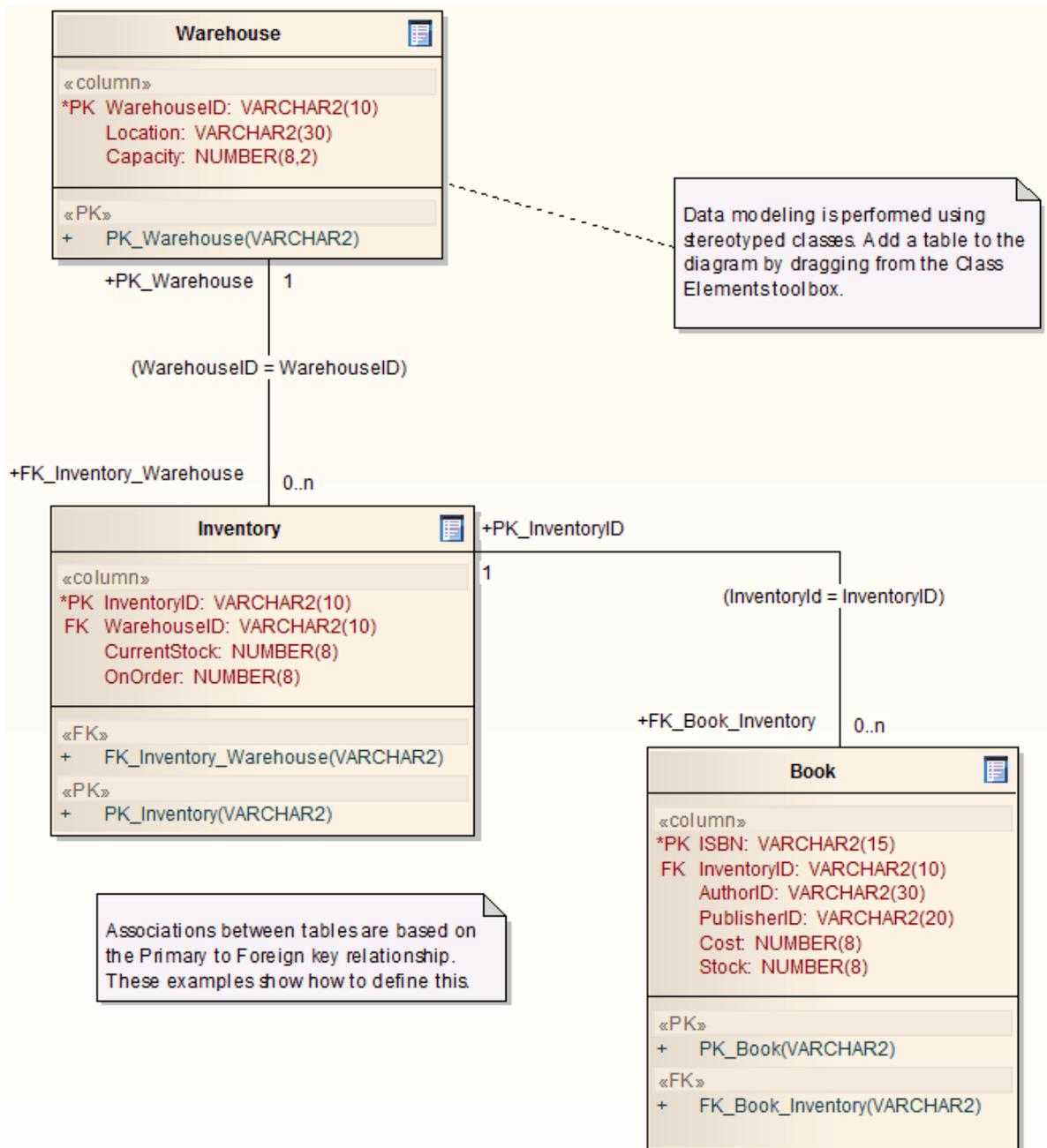
To create a Foreign Key, follow the steps below:

Step	Action	See Also
1	Locate the required Tables in a diagram Both tables must have defined database types	Set The Database Type ^[1348]
2	Select an <i>Associate</i> connector in the Class Relationships page of the Toolbox	
3	Click on the Table to contain the Foreign Key (source) and draw the connector to the other Table (target)	
4	Right-click on the connector and select the Foreign Keys context menu option The Foreign Key Constraint dialog displays	

Step	Action	See Also
5	The default foreign key name is set by the Foreign Key Name Template To change the name to something other than the default provided by the template, select the Override Template checkbox and edit the foreign key name	Define Foreign Key Name Template <small>1365</small>
6	In the Source: panel and the Target: panel, click on the name of each of the two columns involved in the Foreign Key relationship	
7	From the On Delete and/or On Update combo boxes, select the appropriate referential integrity constraint	
8	Click on the Apply or OK buttons to automatically generate the Foreign Key operations	

Example:

The example below shows how a Foreign Key looks in a diagram:

**Notes:**

- For MySQL databases, Foreign Keys are supported for InnoDB tables only.

InnoDB is the default MySQL storage engine as of version 5.5; if you are modeling a MySQL database that is earlier than version 5.5 and you want to use Foreign Keys, you must ensure the **Engine** Tagged Value is set to **InnoDB**

Learn More:

- [Set MySQL Options](#) ^[1349]

10.1.3.2.2 Composite Foreign Key

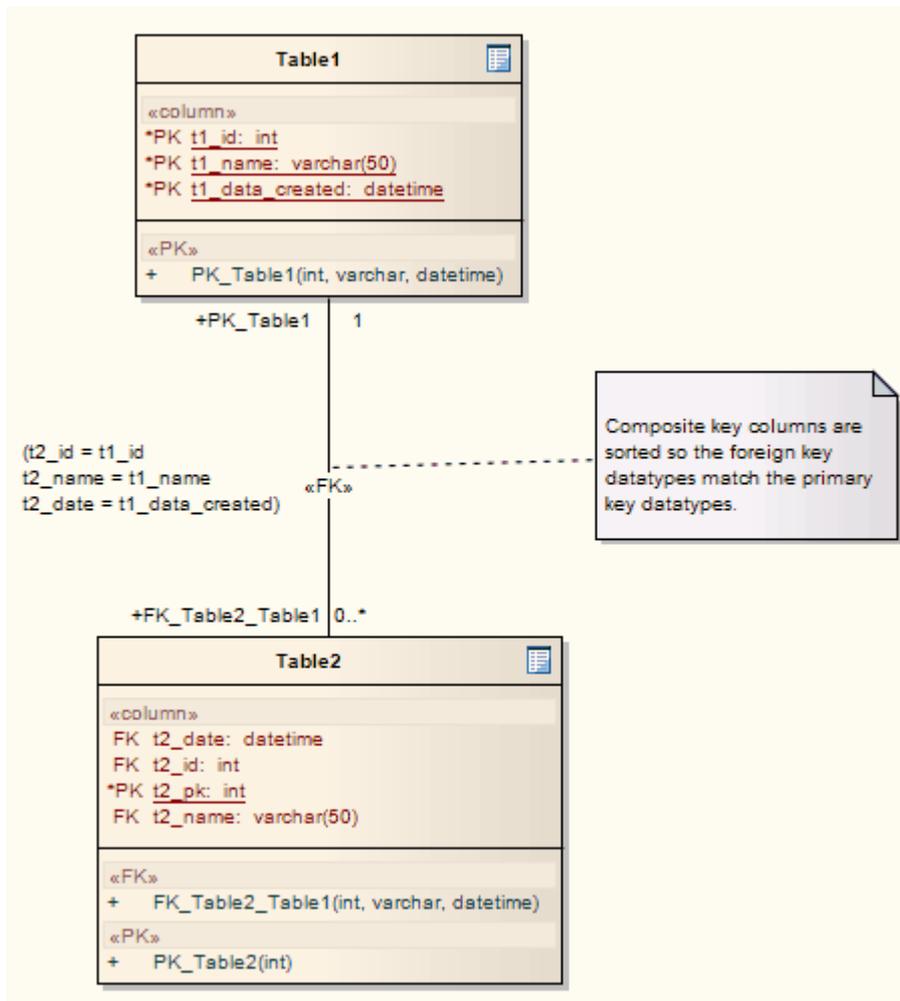
How To:

To create a composite Foreign Key, follow the steps below:

Step	Action	See Also
1	Locate the required Tables in a diagram Both tables must have defined database types	Set The Database Type ^[1348]
2	Select an <i>Associate</i> connector in the Class Relationships page of the Toolbox	
3	Click on the Table to contain the Foreign Key (source) and draw the connector to the other Table (target)	
4	Right-click on the connector and select the Foreign Keys context menu option The Foreign Key Constraint dialog displays	
5	The default foreign key name is set by the Foreign Key Name Template To change the name to something other than the default provided by the template, select the Override Template checkbox and edit the foreign key name	Define Foreign Key Name Template ^[1365]
6	In the Source: panel and the Target: panel, click on the names of the required columns in each panel	
7	Click on the OK button The Foreign Key columns are sorted according to data type to match the data types of the targeted composite Primary Key	
8	If required, you can change the order of the key columns by clicking on the  and  buttons	

Example:

The example below shows how a Composite Foreign Key looks in a diagram:



10.1.3.2.3 Define Foreign Key Name Template

Enterprise Architect enables you to customize the way Foreign Key relationships are named by default. This is helpful if you have specific modeling conventions that stipulate how Foreign Keys are documented.

By changing the **Foreign Key Name Template**, you adjust the name that Enterprise Architect assigns to the Association connector and the Operation that define the Foreign Key relationship.

Access: [Tools | Options | Source Code Engineering | Code Editors: DDL Name Template](#)

How To:

To define the name template for a Foreign Key, follow the steps below:

Step	Action	See Also
1	Select the Code Editors menu option The Code Editors page of the Options dialog displays	
2	Click on the DDL Name Template button	

Step	Action	See Also
	The DDL Name Template dialog displays, showing the default name templates	
3	<p>Edit or replace the name template in the Foreign Key Name Template field</p> <p>For example, if you want to display the Foreign Key description as</p> <p><i>FK_foreigntablename_FKcolumnname_primarytablename_PKcolumnname</i></p> <p>then change the Foreign Key Name Template field to</p> <p><i>FK_%foreigntablename%_%fkcolumnname%_%primarytablename%_%pkcolumnname%.</i></p>	
4	Click on the Save button.	

10.1.4 Stored Procedures

Topics:

Topic	Detail	See also
What is a Stored Procedure?	<p>A stored procedure is a group of SQL statements that form a logical unit and perform a particular task. Stored procedures are used to encapsulate a set of operations or queries to execute on a database server. You can compile and execute stored procedures with different parameters and results, and they can have any combination of input, output and input/output parameters.</p> <p>Enterprise Architect models stored procedures using stereotyped UML Classes and Operations.</p>	

Learn more

- [Create a Stored Procedure](#)^[1367]

10.1.5 Indexes

Topics:

Topic	Detail	See also
What is an Index?	<p>An index is a sorted look-up for a table. When it is known in advance that a table must be sorted in a specific order, it is usually worth the small processing overhead to always maintain a sorted look-up list rather than sort the table every time it is required. In Enterprise Architect, an index is modeled as a stereotyped operation. On generating DDL, the necessary instructions for generating indexes are written to the DDL output.</p>	

Learn more

- [Create an Index](#)^[1367]

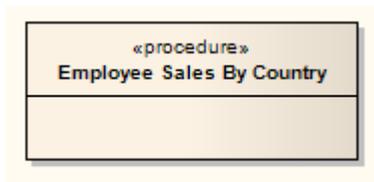
10.1.5.1 Create a Stored Procedure

How To:

To create a stored procedure as a UML Class, follow the steps below.

Step	Action	See Also
1	Open the required diagram.	
2	From the Data Modeling page of the Toolbox (More tools Data Modeling) drag the Procedure icon onto the diagram.	
3	If the Properties dialog does not automatically display, double-click on the element.	
4	In the Database field click on the drop-down arrow and select the target DBMS to model The field displays the default database if it has already been set	
5	In the Procedure definition field, type the entire procedure text.	
6	Click on the OK button.	
7	To rename the stored procedure, select the element, press (F2) and type the new name.	

Example:



10.1.5.2 Create an Index

How To:

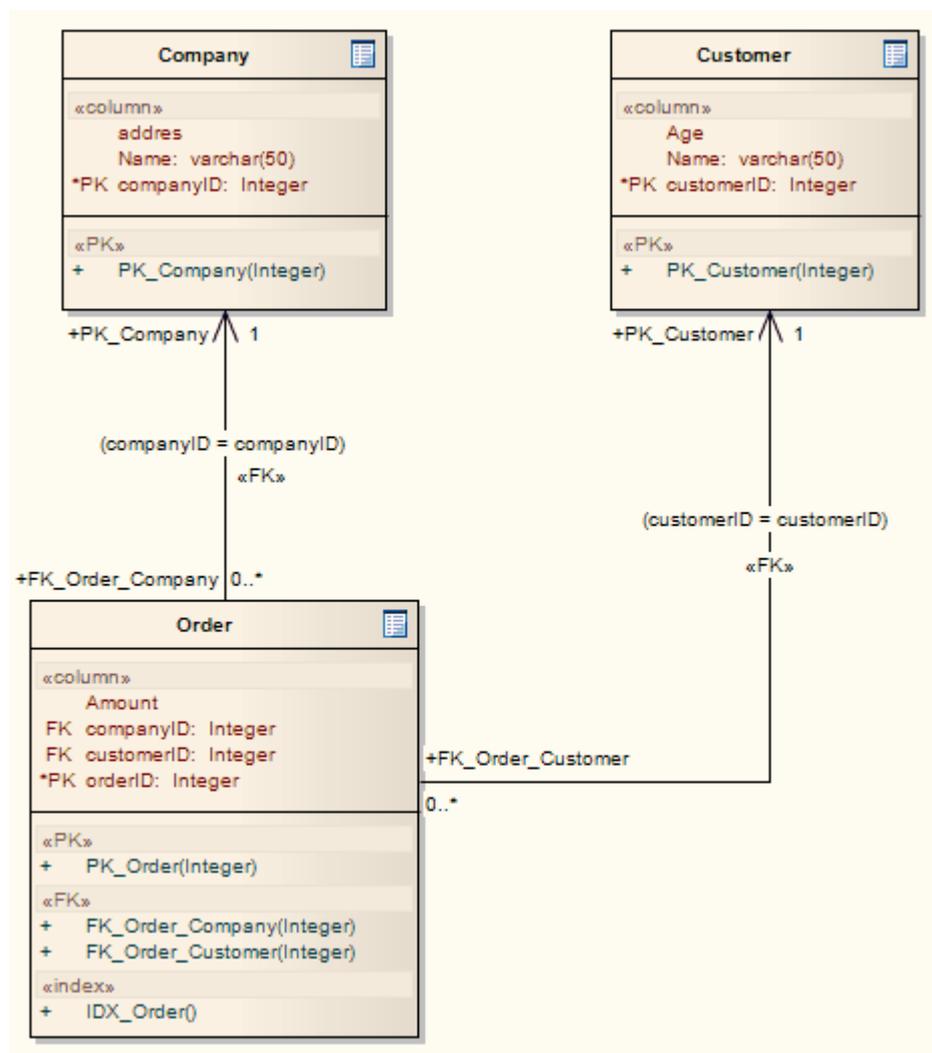
To create an Index, follow the steps below:

Step	Action	See Also
1	Ensure that the column(s) to be used in the index have already been defined in the table.	
2	Right-click on the required table either in a diagram or in the Project Browser .	
3	Select the Operations context menu option The Operations dialog displays.	
4	Add an operation with a name such as <i>IDX_CustomerID</i> . The <i>IDX_</i> prefix is optional, but helps to identify the operation.	

Step	Action	See Also
5	In the Stereotype field for the operation, select index (check and unique are also supported).	
6	Click on the Column tab.	
7	Select the required columns from the Columns drop-down list in the required order	
8	Click the Save button to save changes.	

Example:

In the example below, the *Order* element shows how an index looks in a diagram:



Notes:

- The **unique** characteristic applied to a single column ensures that no two data values in the column can be identical. The **unique** stereotype applied to an index ensures that no two combinations of values across a set of columns can be identical

Learn more:

- [Define Columns](#)^[1357]

10.1.6 Triggers

Topics:

Topic	Detail	See also
What is a Trigger?	A trigger is an operation automatically executed as a result of the modification of data in the database, and usually ensures consistent behavior of the database. For example, a trigger might be used to define validations that must be performed every time a value is modified, or might perform deletions in a secondary table when a record in the primary table is deleted. In Enterprise Architect, a trigger is modeled as a stereotyped operation.	

Learn more

- [Create a Trigger](#)^[1369]

10.1.6.1 Create a Trigger

How To:

To create a Trigger, follow the steps below

Step	Action	See Also
1	Locate the required table in either a diagram or the Project Browser	
2	Use the context menu to open the Operations dialog	
3	Add an operation with a name such as <i>TRG_OnCustomerUpdate</i> The <i>TRG_</i> prefix is optional, but helps to identify the operation	
4	In the Stereotype field select trigger	
5	Click the Save button	
6	Select the trigger operation, then the Behavior tab	
7	In the Initial Code field, enter the entire trigger code including the <i>CREATE_TRIGGER</i> statement	
8	Click the Save button	

10.1.7 Check Constraints

Topics:

Topic	Detail	See also
What is a Check Constraint?	A <i>Check Constraint</i> enforces domain integrity by limiting the values that are accepted by a column.	

Learn more:

- [Create a Check Constraint](#)^[1370]

10.1.7.1 Create a Check Constraint

How To:

To create a Constraint, follow the steps below:

Step	Action	See Also
1	Locate the required table in either a diagram or the Project Browser	
2	Use the context menu to open the Operations dialog	
3	Add an operation with a name such as <i>CHK_ColumnName</i> The <i>CHK_</i> prefix is optional, but helps to identify the operation	
4	In the Stereotype field select check	
5	Click the Save button	
6	Select the check constraint operation, then the Behavior tab	
7	In the Initial Code field, enter the entire check constraint clause (for example, col1 < 1000)	
8	Click on the Save button	

10.1.8 Views

Topics:

Topic	Detail	See also
What is a Database View?	A Database View defines a subset of the database, aggregated into a dynamic, 'virtual' table'. The information presented by a Database View is not physically replicated, rather it is automatically derived based on the query that defines the view. Enterprise Architect models Database Views as stereotyped UML Classes.	

Learn more

- [Create a View](#)^[1371]

10.1.8.1 Create a View

How To:

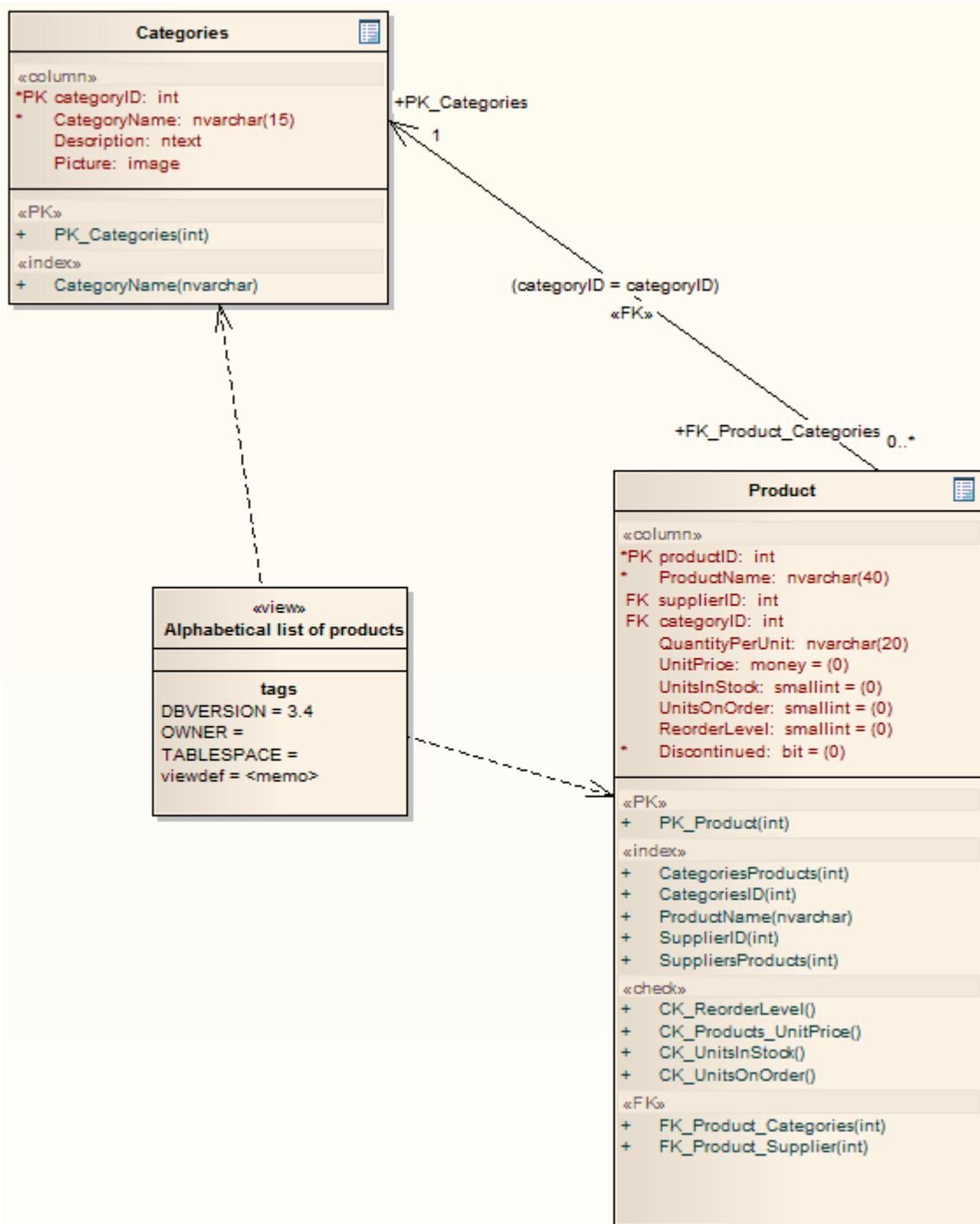
To create a database View, follow the steps below:

Step	Action	See Also
1	On the Data Modeling page of the Toolbox (More tools Data Modeling) , drag the View icon onto your Data Modeling diagram	
2	If the View Properties dialog does not immediately display, double-click on the element	
3	From the Database drop-down list, select the target DBMS to model. The default database displays if it has already been set	
4	Click on the OK button	
5	To rename the View, select the element, press (F2) and type the new name	

To create the View definition, follow the steps below

Step	Action	See Also
1	Create a Dependency connector from the View to the table or tables on which the View depends	
2	Double-click on the View to display the Properties dialog The tables are now listed in the Dependencies field	
3	In the View definition field , type the full view definition The code editor provides intellisense for basic SQL keywords and functions	Code Editor ^[1394]
4	Click on the OK button to save your definition	

Example:

**Notes:**

- The View definition and certain other parameters are held as Tagged Values. The View definition is held in the *viewdef* memo Tagged Value, as shown in the following example diagram. You can select and view the *viewdef* Tagged Value in the **Tagged Values** window, and include it in RTF reports by inserting the **valueOf(viewdef)** field in the *Package::Element* or *Element::Tagged Values* sections

Learn More:

- [Tagged Values](#) ⁷⁶²

- [valueOf\(viewdef\)](#)^[1743]

10.1.9 Oracle Packages

How To:

To create an Oracle package, follow the steps below:

Step	Action	See Also
1	Add a Class element to your data model.	
2	Open the Properties dialog for the element and, in the Stereotype field, type the value Package .	
3	For the package specification, create an Operation with the name <i>Specification</i> and with no return type.	
4	Open the Properties dialog for the <i>Specification Operation</i> and, on the Behavior tab, type the entire package specification into the Initial Code field.	
5	For the package body, create an Operation with the name <i>Body</i> and with no return type.	
6	Open the Properties dialog for the <i>Body Operation</i> and, on the Behavior tab, type the entire package body into the Initial Code field.	

Learn more

- [How to Use the Initial Code Field](#)^[711]

10.1.10 Default Constraints

Default Constraints only apply to SQL Server, and are supported to maintain compatibility with earlier versions.

Access: **Table context menu | Operations**

Use to:

- Create a default constraint

How to:

To create a Default Constraint using the Operations Properties dialog, follow the steps below:

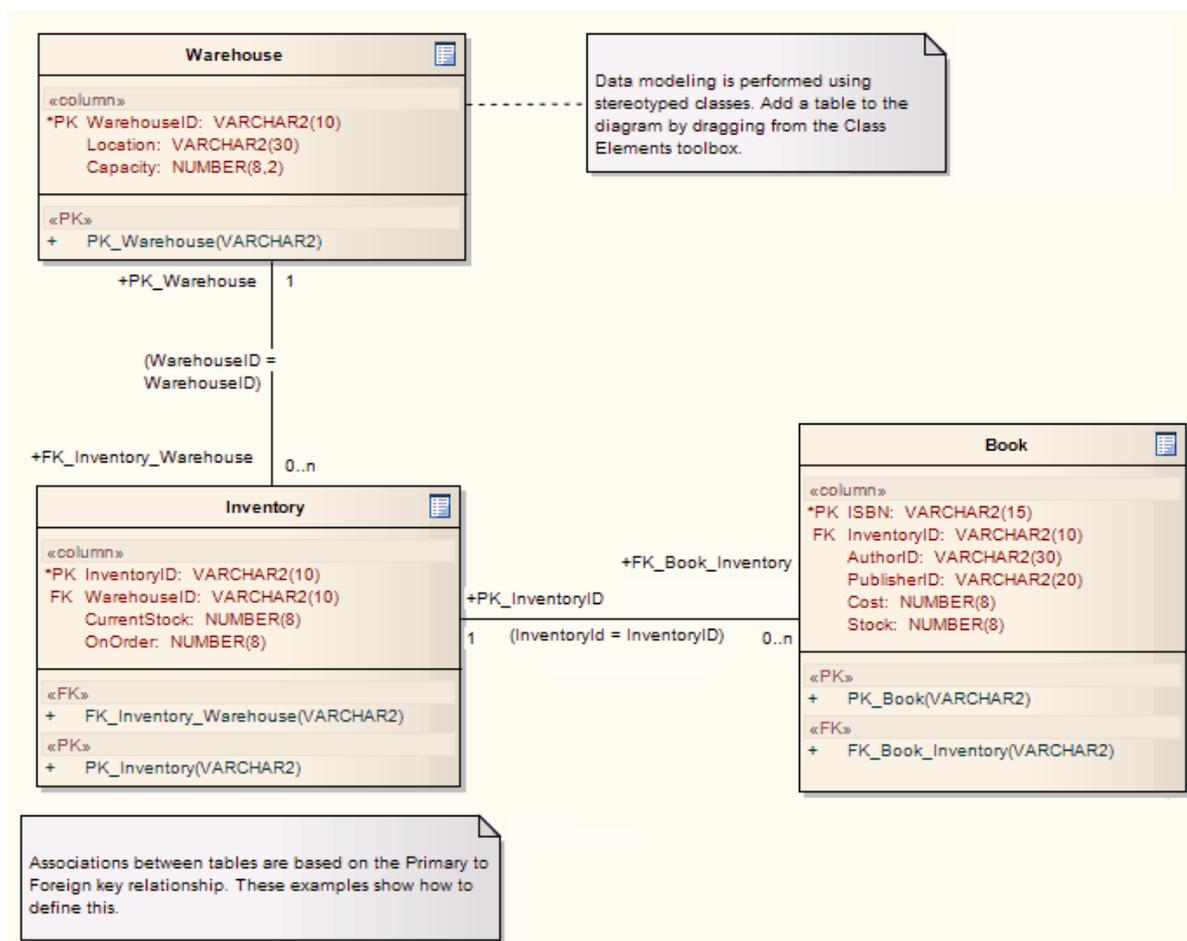
Step	Action
1	In the Name field, type the new operation name DF_<ColumnName> The DF_ is optional, but helps to identify the operation
2	In the Stereotype field type or select the value default
3	Click on the Save button

Step	Action
4	In the Operations list, click on the default operation name, then click Column in the left panel
5	Select the appropriate column from the Column Name drop down list
6	Click on Behavior in the left-hand panel
7	In the Initial Code field, type the default value
8	Click on the Save button

10.1.11 A Data Model Diagram

An example of a *Data Model* diagram is provided below, showing three tables that are linked on *primary to foreign* key pairs with associated multiplicity.

Example:



10.2 Import Database Schema

Analysis of legacy database systems is possible using Enterprise Architect's reverse engineering capabilities. By connecting to a live database via ODBC, you can import the database schema into a standard UML model. Subsequent imports enable you to maintain synchronization between the data model and the live database.

Enterprise Architect supports importing database tables from an ODBC data source. Tables are imported as stereotyped Classes with suitable data definitions for the source DBMS.

How To:

To import database schema and objects, follow the steps below

Step	Action	See Also
1	Select any package in the Logical View	
2	<p>To import into:</p> <ul style="list-style-type: none"> The package only, right-click on the package to display the context menu, and select the Code Engineering Import DB Schema from ODBC menu option A diagram, right-click on the diagram in the selected package to open the context menu, and select the Import DB schema from ODBC menu option <p>The Import DB Schema from ODBC Source dialog displays</p>	
3	<p>In the Database field, click on the (...) (Browse) button and select a suitable ODBC data source from the ODBC dialog (ODBC must be installed and configured on your machine for this to work correctly)</p> <p>When you have selected the data source, the Database field shows the DBMS, the database server ID and the database name, separated by full stops; that is:</p> <p><i>dbms.dbserver.database.</i></p>	Select a Suitable Data Source ^[1377]
4	<p>You can filter objects to be retrieved from the database by schema or owner</p> <p>In the Schema/Owner field, type the schema/owner as a single entry, or as a comma-separated list</p> <p>The filter is useful for databases that support multiple schemas or owners, such as SQL Server 2005/2008, Oracle, PostgreSQL and DB2 Express</p> <p>By default, when importing database objects from Oracle, Enterprise Architect inserts the Oracle user name in this field to avoid unnecessary retrieval of large numbers of objects not owned by that user</p>	
5	<p>In the Filter panel, select the appropriate checkboxes for additional items to include in the import</p> <p>Select the appropriate checkboxes to import system tables and views, user views, triggers and/or Oracle packages</p> <p>If you select to import User Functions and/or User Sequences as individual Classes, then they are imported as separate elements and the Properties dialog is solely concerned with the Function or Sequence definition; for Stored Procedures, always select this option</p>	

Step	Action	See Also
	If you select to import User Functions and/or User Sequences as Class operations, then they are imported as operations (methods) and you view and edit them through the Operations Properties dialog of the parent Class	
6	In the Synchronization panel, select the appropriate option to determine whether the existing Classes are to be synchronized, or the database objects imported as new objects If you select the Synchronize existing classes option, also select the appropriate checkboxes to determine whether model comments, column default values and/or table constraints are to be retained or overwritten with the comments, values and constraints of the imported objects	
7	Click on the Import button to start the import	
8	Select the database objects to import	Select Tables ^[1377] Imported Class Elements ^[1377]

Notes:

- Import of stored procedures and views is supported for: DB2; SQL Server; Firebird/Interbase; Informix; Ingres; Oracle 9i, 10g and 11g; MySQL; SQLite, PostgreSQL; Sybase Adaptive Server Enterprise (ASE) and Sybase Adaptive Server Anywhere (ASA)
- If you are importing database schema from an MS Access Jet 4.0 database, please ensure that you have selected the **Use Jet 4.0** checkbox on the General page of the Options dialog, otherwise the Jet 3.5 routines are loaded; you must restart Enterprise Architect after selecting the checkbox
- Enterprise Architect requires **32-bit ODBC** drivers to connect to a repository through ODBC; to set up the ODBC configuration on **64-bit** clients, run the 32-bit ODBC Data Source Administrator from C: \Windows\SysWOW64\odbcad32.exe
- The ODBC connection should use the ODBC driver available from the DBMS vendor, such as MySQL's ODBC driver for MySQL, and Oracle's ODBC driver for Oracle; drivers provided by third-party vendors are not supported, including the Microsoft ODBC driver for Oracle
- If setting up a ODBC connection for reverse engineering, the default settings are sufficient
- Additional data types are available from the Datamodeling Data Types section of the Resources page on the Sparx Systems website
- For importation, you can select the **Tools | Database Engineering | Import DB Schema from ODBC** menu option as an alternative
- It is only possible to import into a diagram if it is in the selected package - if a diagram from another package is open, a message displays to give the option to cancel the import or to continue importing into the package only; the Import DB Schema from ODBC Source dialog includes checkbox options to import into the diagram and package, or into the package only
- If no diagram is open, the **Package Only** radio button defaults to selected and the options are disabled; if the open diagram is in the selected package, you can select either option

Learn More:

- [Import Source Code](#) ^[1508]
- [General Settings](#) ^[422]
- [Data Modeling Data Types](#) (Online Resource)

10.2.1 Select a Data Source

Topics:

Topic	Detail	See also
General Usage	<p>To import DDL from existing data sources, you must have a suitable ODBC connection installed and configured</p> <p>From the Import DB Schema from ODBC Source dialog you can select the ODBC data source using the standard windows ODBC set-up dialog; click on the data source name and then click on the OK button</p>	

Learn More:

- [Connect to a Data Repository](#)^[16]

10.2.2 Select Tables

When you have opened the ODBC data source, Enterprise Architect acquires a list of database objects suitable for importing. This is presented in a list form for you to select from.

Highlight the schema to import from the top list. Alternatively, select individual objects from the lower list.

Topics:

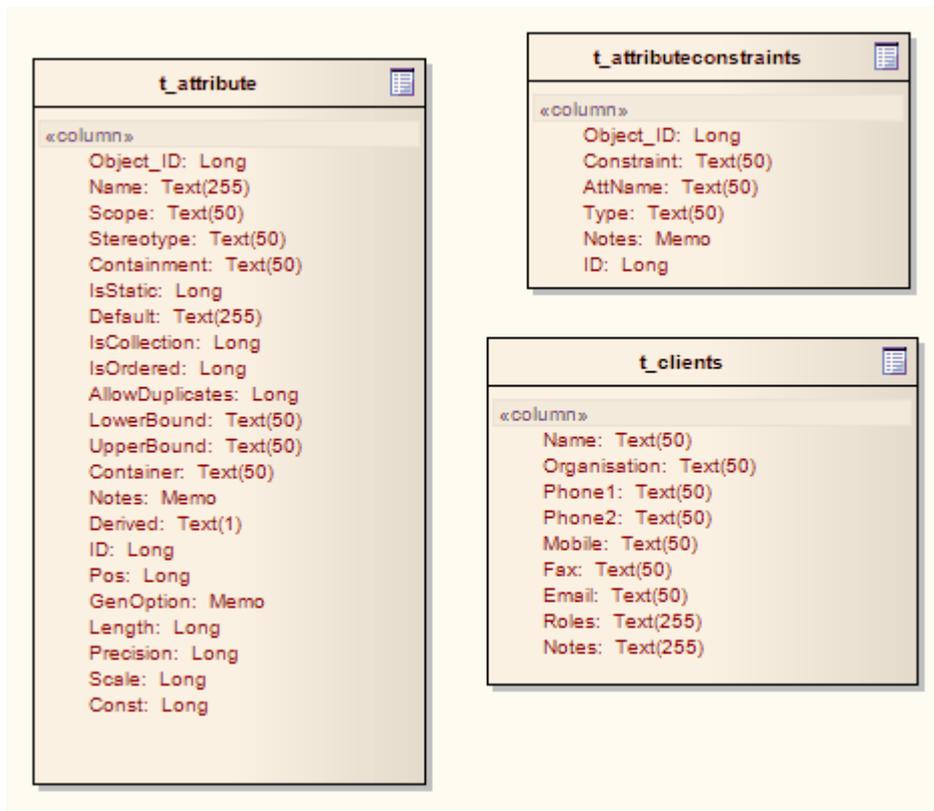
Topic	Detail	See also
Selection Shortcuts	<ul style="list-style-type: none"> • To select all objects, click on the Select All button • To clear all objects, click on the Select None button • Hold down (Ctrl) while clicking to select multiple objects • Hold down (Shift) and click on the first and last objects in a range to select that range <p>When you have selected the objects, click on the OK button</p>	

10.2.3 The Imported Class Elements

When you import DDL table definitions they are converted to stereotyped Classes according the *UML Data Modeling Profile*.

The image below shows some example tables imported into the model using an ODBC data connection.

Example:



10.3 Generate DDL

Enterprise Architect enables you to generate DDL from your model for Tables and Packages.

Learn More:

- [Generate DDL for Tables](#) ^[1379]
- [Generate DDL for Packages](#) ^[1380]

10.3.1 Generate DDL For a Table

How To:

To generate DDL for a table, follow the steps below:

Step	Action	See Also
1	In the diagram, right-click on the table for which to generate DDL The context menu displays	
2	Select the Generate DDL option The Generate DDL dialog displays	
3	In the Path field, use the (...) (Browse) button to select the filename of the script to create	
4	To include comments in the DDL, in the Comment Level field select the appropriate level For example, Column for comments on columns, or All for comments on all structures	
5	Select the checkboxes for the appropriate inclusions; for example, to include a 'drop table' command in the script, select the Create Drop SQL checkbox Deselect the checkboxes for inclusions you do not require	
6	To create the DDL, click on the Generate button	
7	To view the output, click on the View button (you must configure a DDL viewer in the Local Settings dialog first)	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Source Code and DDL** permission to generate DDL
- Some checkboxes display only if the appropriate database is defined for the table; for example, **IF EXISTS** displays only if the database for the table is PostgreSQL, and **SEQ_** and **_SEQ** radio buttons display only if the database for the table is Oracle
- For a PostgreSQL database, you must select the **Generate Sequences** checkbox to enable auto increment columns to be created
- If generating Oracle sequences, you must always select the **Generate Triggers** and **Generate Sequences** checkboxes - this ensures that a pre-insert trigger is generated to select the next sequence value to populate the column; also set the **AutoNum** property to **True** in the column properties
- If generating Oracle sequences, to generate the sequence name and trigger name with the syntax

SEQ_<sequence_name> and *TRG_<trigger_name>*, select the **SEQ_** option; conversely, to generate them with the syntax *<sequence_name>_SEQ* and *SET_<trigger_name>*, select the **_SEQ** option

Learn More:

- [Permission List](#)^[205]
- [Create Columns](#)^[135]

10.3.2 Generate DDL for a Package

In this procedure, you can generate DDL for a package, and also compare the DDL with the database.

Access: **Project Browser package context menu | Code Engineering | Generate DDL**
Select package Tools | Database Engineering | Generate Package DDL

How To:

To generate DDL for a package, follow the steps below:

Step	Action	See Also
1	Select the menu option The Generate Package DDL dialog displays	
2	Select the checkbox against each inclusion required Deselect the checkboxes for inclusions you do not require	
3	To recursively generate DDL, select the Include All Child Packages checkbox	
4	Select the Save Generated Order checkbox to save the order in which the objects are generated This is useful where the order is changed to resolve object dependencies; the order is saved when you click on the Generate button	
5	If you click the Refresh button with the Save Generated Order checkbox <i>unchecked</i> , the objects are restored to their original order	
6	Click on the Generate button to proceed Enterprise Architect prompts you for file names as the process executes	

To compare the generated DDL with the database, follow the steps below

Step	Action	See Also
1	On the Generate Package DDL dialog, click on the Compare button The Compare With Database dialog displays	
2	Click on the (...) button and locate the required database on the Select Data Source dialog	
3	For an Oracle database, if required you can also specify the Owner in the Schema/Owner field	

Step	Action	See Also
4	Click on the View button to perform the comparison The Comparison Database dialog displays with the results of the comparison Click on each table name to review information on that table	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Source Code and DDL** permission to generate DDL
- Some checkboxes display only if the appropriate database is defined for the table. For example, **IF EXISTS** displays only if the database for the table is PostgreSQL, and **SEQ_** and **_SEQ** radio buttons display only if the database for the table is Oracle
- For a PostgreSQL database, you must select the **Generate Sequences** checkbox to enable auto increment columns to be created
- If generating Oracle sequences, you must always select the **Generate Triggers** and **Generate Sequences** checkboxes - this ensures that a pre-insert trigger is generated to select the next sequence value to populate the column; also set the **AutoNum** property to **True** in the column properties
- If generating Oracle sequences, to generate the sequence name and trigger name with the syntax *SEQ_<sequence_name>* and *TRG_<trigger_name>*, select the **SEQ_** option; conversely, to generate them with the syntax *<sequence_name>_SEQ* and *SET_<trigger_name>*, select the **_SEQ** option

Learn More:

- [Permission List](#)^[205]
- [Create Columns](#)^[135]

10.4 Entity Relationship Diagrams (ERDs)

According to the online [Wikipedia](#):

An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called Entity-Relationship Diagrams, ER Diagrams, or ERDs.

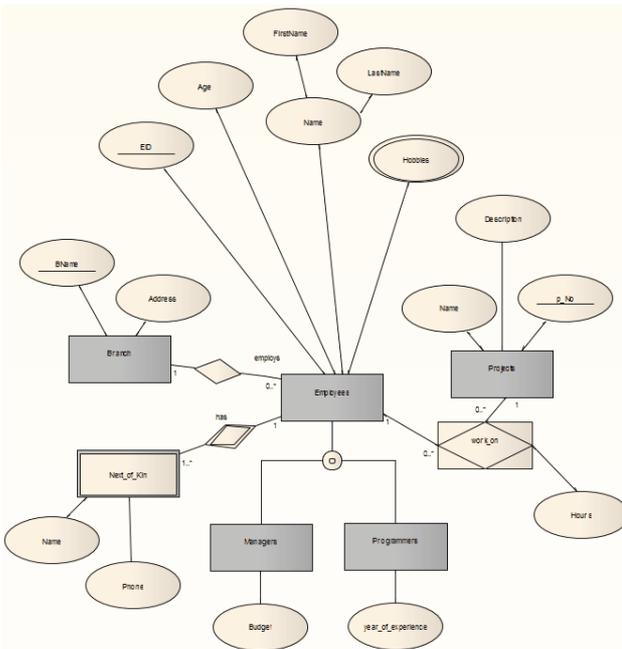
Entity Relationship Diagrams in Enterprise Architect:

Entity Relationship Diagrams in Enterprise Architect are based on Chen's ERD building blocks: entities are represented as rectangles, attributes are represented as ellipses and relationships are represented as diamond-shape connectors.

ERD technology in Enterprise Architect assists you in every stage from building conceptual data models to generating Data Definition Language (DDL) for the target DBMS.

Topics:

Topic	Detail	See also
MDG Technology for ERD and ERD Transformations	<p>Enterprise Architect enables you to develop Entity Relationship diagrams quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The Entity Relationship diagram facilities are provided in the form of:</p> <ul style="list-style-type: none"> • An Entity Relationship diagram type, accessed through the New Diagram dialog • An Entity Relationship Diagram page in the Toolbox • Entity Relationship element and relationship entries in the Toolbox Shortcut Menu and Quick Linker <p>Enterprise Architect also provides transformation templates to transform Entity Relationship Diagrams into Data Modeling Diagrams, and vice versa.</p>	<p>New Diagram Dialog^[568]</p> <p>Toolbox Shortcut^[557]</p> <p>Quick Linker^[622]</p> <p>ERD to Data Model Transformation^[1317]</p> <p>Data Model to ERD Transformation^[1310]</p>
Entity Relationship Diagram Toolbox Page 	<p>You can access the Entity Relationship Diagram page of the Toolbox through the More tools Entity Relationship Diagrams menu option.</p> <ul style="list-style-type: none"> • Entity is an object or concept that is uniquely identifiable. The property of Multiplicity in the SourceRole and TargetRole definitions for the Relationship connector (below) can be used to define the cardinality of an Entity that participates in this relationship. • Attribute is a property of an entity or a relationship type • N-ary Association represents unary (many-to-many recursive) or ternary relationships and can also be used to represent relationships that have attributes among the entities; Note that the N-ary Association element should always be at the target end of a connector • Connector is a connector between an Entity and an 	

Topic	Detail	See also
	<p>Attribute, and between two Attributes</p> <ul style="list-style-type: none"> Relationship is a diamond-shape connector, representing the meaningful association among entities. Disjoint and Overlapping represent the relationships between the super-class Entity and the sub-class Entity 	
<p>A typical Entity Relationship Diagram</p>	 <p>The diagram illustrates a central 'Employee' entity with attributes: ED (underlined), Age, Name (with sub-attributes FirstName and LastName), and Hobbies. It is connected to several other entities: <ul style="list-style-type: none"> 'Branch' (1 to 1) via an 'employs' relationship (diamond connector). 'Project' (0..1 to 1) via a 'works_on' relationship (diamond connector). 'Managers' (1 to 1) via a 'has' relationship (diamond connector). 'Programmers' (1 to 1) via a 'has' relationship (diamond connector). 'Next_of_Kin' (1..1 to 1) via a 'has' relationship (diamond connector). Other entities and their attributes: <ul style="list-style-type: none"> 'Branch': Name (underlined), Address. 'Project': Name, D_No (underlined), Description. 'Managers': Budget. 'Programmers': year_of_experience. 'Next_of_Kin': Name, Phone. 'works_on' relationship: Hours. </p>	
<p>Disable Entity Relationship Diagrams</p>	<p>If you prefer not to use Entity Relationship Diagrams in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	<p>MDG Technologies 1033</p>

Tagged Values:

Some of the Entity Relationship diagram components can be modified by Tagged Values, as indicated below:

Component	Tagged Value	Notes
Entity	isWeakEntity	If true, this entity is a weak entity.
Attribute	attributeType	Four options: <ul style="list-style-type: none"> normal Attribute primary key attribute multi-valued Attribute derived Attribute
	commonData	Defines the common data type for each attribute.

Component	Tagged Value	Notes
	taType	
	dbmsDataType	<p>Defines the customized DBMS data type for each attribute.</p> <p>You must define the customized type first through the Settings Database Datatypes menu option</p> <p>Also, set the <i>commonDataType</i> tag to na to activate the <i>dbmsDataType</i> tag.</p>
N-ary Association	isRecursive	<p>If true, the N-ary Association represents the many-to-many recursive relationship.</p> <p>For one-to-many and one-to-one recursive relationships, we suggest using the normal Relationship connector.</p> <p>Sometimes you might want to limit the stretch of the diamond-shape Relationship connectors. Simply pick a Relationship connector, right-click to display the context menu, and select the Bend Line at Cursor option.</p>
Relationship	isWeak	If true, the Relationship is a weak relationship.
Disjoin Overlapping	Participation	Two options: partial and total.

Notes:

- Entity Relationship Diagrams are supported in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

Part



11 Software Engineering



Software Engineering is the process of designing, implementing and maintaining software. This section describes how Enterprise Architect can assist in this process by providing a design environment and tools including **automated code generation**, **reverse engineering** of source code and **synchronization** between the source code and model.

Topics:

Topic	Link
Software Design	GoF Patterns ^[1545]
Development Environment	Development Tools ^[1387] Code, Build & Debug ^[1390] MDG Integration and Code Engineering ^[1514]
Generate Source Code	Generate Source Code ^[1490]
Importing Source Code	Importing Source Code ^[1508]

Learn More:

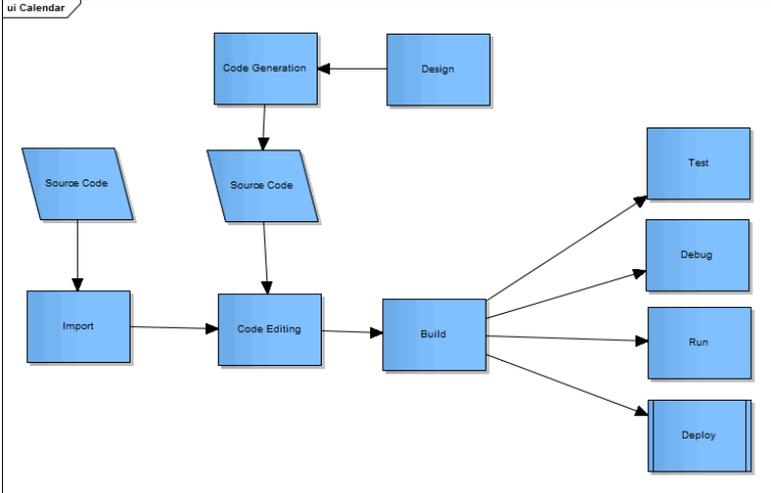
- Enterprise Architect provides a set of tools that goes beyond traditional debugging capabilities, see [Visual Execution Analyzer](#) ^[1635]
- [Database Engineering](#) ^[1345] is also available, including keys, triggers, constraints, RI and other relational database features, for and from a range of database products
- Enterprise Architect provides technologies to support Service Oriented Architecture; see [SOA and XML Engineering](#) ^[1580]

11.1 Development Tools



Topics:

Topic	Detail	See also
A Tightly Integrated Development Environment with Outstanding Tools and Functionality	<ul style="list-style-type: none"> • Design and elaborate your Class model in UML and other technologies • Generate code and reverse engineer existing code • Customize code generation with templates • Invoke compilers and other command line tools to build, run and deploy • Use the built in source code editor to write and maintain source files • Use the built in debugger to inspect and correct code at run time • Use the Profiler to fine tune Windows native C/C++ code • Use the visualization tools to generate sequence diagrams from executing code • Use the testing capabilities to validate execution with class and method level constraints and invariants • Link to junit and nUnit test cases • Link or integrate with Eclipse or Visual Studio when necessary to further enhance your development process 	
Abstract	<p>Enterprise Architect (Professional edition and above) provides an extraordinary range of features and a unique, rich, tightly coupled toolset for rapidly developing, visualizing, debugging, testing, maintaining and generally working with even the most complex source code</p> <p>A wide range of standard programming languages are supported for design and reverse engineering; inbuilt editors with custom syntax highlighting and tight integration with the model provide a powerful and effective programming environment; coupled with the build/debug and testing capabilities, Enterprise Architect is both a remarkable analysis and design tool, and a formidable debug and testing environment</p> <p>Enterprise Architect's Visual Execution Analyzer also supports Java, .NET and native Windows applications written in C/C++ using Microsoft's compilers; in addition to powerful tools for standard debugging, Enterprise Architect supports recording of Sequence diagrams from executing code, capturing of method calls and conversion into custom call sets for testing purposes, validation of execution against State diagrams, and more</p> <p>The Model Driven Development Environment (MDDE) provides one of the richest and most useful feature sets available for working with source code and ensuring the construction of robust and effective applications; the MDDE integrates code and model by providing</p>	

Topic	Detail	See also
	options to either generate source code from the model or reverse engineer existing source code into a model - source code and model can be synchronized in either direction	
Relationships of Software Engineering Units	 <pre> graph TD Design[Design] --> CG[Code Generation] CG --> SC1[/Source Code/] SC1 --> Import[Import] SC2[/Source Code/] --> CE[Code Editing] Import --> CE CE --> Build[Build] Build --> Test[Test] Build --> Debug[Debug] Build --> Run[Run] Build --> Deploy[Deploy] </pre>	
Supported Languages	<p>Popular languages supported include:</p> <ul style="list-style-type: none"> • C/ C++ • Java • Microsoft .NET family • ADA • Python • Perl <p>Toolboxes provide for different modeling technologies</p>	

Notes:

- Although you can generate and reverse engineer code in a range of languages, Execution Analysis debugging and recording are supported for the following platforms / languages only:
 - Microsoft Windows Native C
 - Microsoft Windows Native C++
 - Microsoft Windows Visual Basic
 - Microsoft .NET Family (C#, J#, VB)
 - Sun Microsystems Java

Learn More:

- [Getting Started](#)^[1391]
- [Basic Setup](#)^[1392]
- [Software Engineering](#)^[1386]
- [Using Code Editors](#)^[1394]
- [Build Application](#)^[1415]
- [Debug](#)^[1441]
- [Test](#)^[1451]
- [Run](#)^[1451]
- [Deploy](#)^[1453]

11.1.1 Overview of Development

Code Engineering with Enterprise Architect broadly encompasses various processes for generating or transforming code from your UML model and importing code into the model, to support model development in several coding languages, database development and SOA development.

Topics:

Topic	Detail	See also
Model Driven Code Engineering	<ul style="list-style-type: none"> Source code generation and reverse engineering for many popular languages, including C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript, Python and PHP A built in 'syntax highlighting' source code editor Code generation templates, which enable you to customize the generated source code to your company specifications 	Software Engineering ^[1386]
Transformations for Rapid Development	<ul style="list-style-type: none"> Advanced Model Driven Architecture (MDA) transformations using transformation templates Built-in transformations for DDL, C#, Java, EJB and XSD One Platform Independent Model can be used to generate and synchronize multiple Platform Specific Models, providing a significant productivity boost 	Model Transformations - MDA ^[1301]
Visual Execution Analysis / Debugging, Verification and Visualization	<ul style="list-style-type: none"> Execute Build, test, debug, run and deploy scripts Integrate UML development and modeling with source development and compilation Generate NUnit and JUnit test Classes from source Classes using MDA Transformations Integrate the test process directly into the Enterprise Architect IDE Debug .NET, Java and Microsoft Native (C, C++ and Visual Basic) applications Design and Execute Test suites based on <i>Programming by Contract</i> principles 	Visual Execution Analysis ^[1635]
Database Modeling	<p>Enterprise Architect enables you to:</p> <ul style="list-style-type: none"> Reverse engineer from many popular DBMSs, including SQL Server, My SQL, Access, PostgreSQL and Oracle 9i, 10g or 11g Model database tables, columns, keys, foreign keys and complex relationships using UML and an inbuilt data modeling profile Forward generate DDL scripts to create target database structures 	Database Engineering ^[1345]
XML Technology Engineering	<p>Enterprise Architect enables you to rapidly model, forward engineer and reverse engineer two key W3C XML technologies:</p> <ul style="list-style-type: none"> XML Schema (XSD) Web Service Definition Language (WSDL) <p>XSD and WSDL support is critical for the development of a complete Service Oriented Architecture (SOA), and the coupling of UML 2.3 and XML provides the natural mechanism for implementing XML-based SOA artifacts within an organization</p>	XML Schema - XSD ^[1581] Web Services - WSDL ^[1611]

11.2 Code, Build & Debug



This section discusses the integrated Model Driven Development Environment (MDDE).

Topics:

Topic	Detail	See also
Abstract	<p>Model Driven Development provides a more robust, accessible and faster development cycle than traditional coding-driven cycles</p> <p>A well constructed model, intimately linked with source code build, run, debug, test and deploy capabilities provides a rich, easily navigated and easily understood target architecture</p> <p>Traceability, linkage to Use Cases, Components and other model artifacts, plus the ability to readily record and document pre-existing or recently developed code, make Enterprise Architect's development environment uniquely powerful</p> <p>Enterprise Architect incorporates industry standard intelligent editing, debuggers and modeling languages</p>	<p>Getting Started ^[1391]</p> <p>Setup ^[1392]</p> <p>Code Editors ^[1394]</p> <p>Build ^[1415]</p> <p>Clean ^[1416]</p> <p>Debugging ^[1416]</p> <p>Run ^[1451]</p> <p>Testing ^[1451]</p> <p>Deploying ^[1453]</p> <p>Searching Files ^[1454]</p>
	<p>The MDDE provides tools to design, build and debug an application:</p> <ul style="list-style-type: none"> • UML technologies and tools to model software • Code generation tools to generate/reverse engineer source code • Tools to import source code and binaries • Code editors that support different programming languages • Intellisense to aid coding • Analyzer scripts that enable a user to describe how to build, debug, test and deploy the application <pre>pApp = new CBCGPAAppointmentDemo { 2 of 2 CBCGPAAppointmentDemo::CBCGPAAppointmentDemo(COleDateTime& dtStart, COleDateTime& dtFinish, CString& strText, COLORREF clrBackground, COLORREF clrForeground, COLORREF clrDuration) RGB (165, 222, 99), CLR_DEFAULT, RGB(128, 0, 128) };</pre>	<p>Development Tools ^[1387]</p> <p>MDG Technologies ^[1037]</p> <p>Modeling Basics ^[513]</p> <p>Generate Source Code ^[1490]</p> <p>Import Source Code ^[1510]</p> <p>Import Binary Module ^[1513]</p> <p>Code Editors ^[1394]</p> <p>Intellisense ^[1399]</p> <p>Managing Analyzer Scripts ^[1392]</p>

11.2.1 Getting Started

To quickly start development in the Model Driven Development Environment, check through the following topics:

- [Prerequisites](#) ^[1391]
- [Available Tools](#) ^[1391]
- [Workspace Layouts](#) ^[1391]
- [General Workflow](#) ^[1391]
- [Code Generation and Synchronization - Safeguards](#) ^[1392]

11.2.1.1 Prerequisites

Before using the Model Driven Development Environment:

- You should be using the correct edition: Enterprise Architect Professional, Corporate or extended editions
- You should be connected to the required model
- Relevant source code should be imported into the model
- An Analyzer Script should be configured
- Required External frameworks should be installed

Learn More:

- [Setup](#) ^[1392]

11.2.1.2 Available Tools

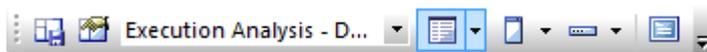
This section describes the tools available in the Model Driven Development Environment:

- [Workspace Layouts](#) ^[1391]
- [Code Engineering](#) ^[1386]
- [Code Editors](#) ^[1394]
- [Intellisense](#) ^[1395]
- [Application Management](#) ^[1392]
- [Debugger Management](#) ^[1416]

11.2.1.3 Workspace Layout

You can choose from many predefined workspace layouts, depending on the tasks you perform. When you are familiar with the environment and controls available to you, you can create your own layouts.

Workspace Toolbar:



Learn More:

- [Workspace Layouts](#) ^[115]

11.2.1.4 General Workflow

In working with the Model Driven Development Environment, you apply the following workflow as a circular process, refining as necessary in each iteration.

- Configure and set up scripts
- Model - Edit - Build - Debug - Test - Profile - Deploy - Document and Analyze

Learn More:

- [Managing Scripts](#)^[1392]

11.2.1.5 Code Generation and Synchronization - Safeguards

It is important that the model and source code are kept synchronized for the Visual Execution Analyzer to produce useful results.

Use the Code Generation tools to synchronize your model after any design changes or code editing.

Always build the application prior to any Execution Analysis session - debugging, recording or profiling.

Learn More:

- [Synchronize Model and Code](#)^[1506]

11.2.2 Setup

The Execution Analyzer is a feature of Enterprise Architect, responsible for building a project, debugging code, recording a Sequence diagram and profiling information from a running program.

In order for these features to work, it is necessary to record information about the application. This is achieved in Enterprise Architect through the use of *Analyzer Scripts*.

An Analyzer Script is always associated with a package. The information you provide in a script controls the actions in, and provides support for, the following tasks:

- [Building](#)^[1415]
- [Debugging](#)^[1416]
- [Testing](#)^[1451]
- [Deployment](#)^[1453]
- [Profiling](#)^[1658]

Learn More:

- [Managing Analyzer Scripts](#)^[1392]
- [Analyzer Script Editor](#)^[1394]

11.2.2.1 Managing Analyzer Scripts

The Execution Analyzer window enables you to manage all scripts in the model, using the window toolbar buttons to control script tasks. Many features, such as Debugging, Testing, Profiling and Simulation, depend on and function according to how the active script has been configured.

Scripts are listed by package; packages that do not have any scripts are not displayed.

Each user can activate one script at any time, independent of other users of the same model; one user activating a script does not impact the currently active scripts for other users or affect the scripts available to them.

Access: **Analyzer | Execution Analyzer (Shift + F12)**

Toolbar Button	Action	See Also
	List the Analyzer windows; select and display the required window	Debugging ^[1416] Breakpoint and Marker Management ^[1436] View the Call Stack ^[1441] View the Local Variables ^[1442] View Variables in Other Scopes ^[1444] Inspect Process Memory ^[1445] The Recording History ^[1651] The Testpoints Window ^[1676] Profiling Native Applications ^[1658] Model Simulation ^[1681]
	Add new Analyzer Script A prompt displays to select the parent package	Analyzer Script Editor ^[1394]
	Export Scripts You can export one or more Analyzer Scripts to an XML file, which can be used to import the scripts into another model; a prompt displays for the target file name and location	
	Import Scripts You can import one or more Analyzer Scripts from a previously exported XML file; a prompt displays for the source file name and location	
	Execute the Debug command of the active script	Setup for Debugging ^[1417]
	Execute the Build command of the active script	Add Commands ^[1415]
	Execute the Test command of the active script	Add Testing Command ^[1451]
	Execute the Run command of the active script	Add Run Command ^[1451]
	Execute the Deploy command of the active script	Add Deploy Command ^[1453]
	Display the Help topic for this window	

11.2.2.2 Analyzer Script Editor

Scripts are associated with a Package. When you create an Analyzer Script you can define a number of actions. The Execution Analyzer Script Editor has a number of pages for configuring various features:

If you plan to use any of the features of the Execution Analyzer, you must complete at least the Build and Debug pages.

Task	Page	Actions	See also
Build	Build	Enter script or command to build the application	Build ^[1415]
	Clean	Enter script or command to clean the previous build	Clean ^[1416]
Test	Test	Enter script or command to test the application	Test ^[1451]
	Testpoints	Specify where the output from a Testpoint run is sent	Testpoint Output ^[1453]
Debug	Platform	Specify the debugging platform, the application to be debugged, and the mode of debugging (attach to process or run)	Debug ^[1416]
	Tracepoints	Specify where the output from Tracepoints encountered during a debug session are sent	Trace Output ^[1435]
	Workbench	Specify required information for using the Object Workbench control	Workbench Setup ^[1663]
Run	Run	Enter a script or command to run the application	Run ^[1451]
Deploy	Deploy	Enter a script or command to deploy the project	Deploy ^[1453]
Recording	Recording	Exclude areas of the application from being recorded; you can specify wildcard expressions to exclude operations, Classes and modules	Configure Recording Detail ^[1639]
Simulation	Simulation	Complete the configuration for Simulation Control	Model Simulation ^[1681]

11.2.3 Code Editors

Enterprise Architect provides a number of editors that you can use across the development environment to maintain scripts, code and templates. Each editor has its own features, but they are all based on a common *Code Editor control*.

Topics:

Topic	Detail	See also
Editors	<p>The provided editors are:</p> <ul style="list-style-type: none"> • Shape Script Editor* • Code Generation Template Editor* • Transformation Template Editor* • Custom SQL Search Filter Editor* • Database View Editor* • HTML Report Template Editor • Source Code Viewer* • Script Editor* <p>You can have several code editors (or files within a code editor) open at the same time, as separate tabs in the central view area of the Enterprise Architect work area</p> <p>You can also close the editors individually or all together, leaving views of other types (such as diagrams or RTF reports) still open</p>	Shape Script Editor ^[1090] Code Generation Template Editor ^[1485] Transformation Template Editor ^[1330] Custom SQL Search Filter Editor ^[484] Database View Editor ^[1371] HTML Report Template Editor ^[1804] Source Code Viewer* ^[1408] Script Editor ^[1407] Diagram Tabs ^[543]
Code Editor	<p>The common Code Editor provides a variety of functions to assist with the code editing process, including:</p> <ul style="list-style-type: none"> • Syntax Highlighting • Bookmarks • Cursor history • Brace matching • Automatic indentation • Commenting selections • Scope guides • Zooming • Line selection • intellisense • Find and Replace • Find in Files <p>A range of these functions is available through keyboard key combinations and/or context menu options</p> <p>You can customize several of the Code Editor features by setting properties in the Code Editor configuration files; for example, by default the line containing the cursor is always highlighted, but you can turn the highlighting off</p> <p>Additional information is provided in the <i>Code Editor Configuration Guide</i>, located as a PDF file in the <i>Config</i> directory under your Enterprise Architect installation directory</p>	Syntax Highlighting ^[1395] Bookmarks ^[1396] Cursor history ^[1396] Brace matching ^[1397] Automatic indentation ^[1397] Commenting selections ^[1397] Scope guides ^[1398] Zooming ^[1398] Line selection ^[1399] intellisense ^[1399] Find and Replace ^[1412] Find in Files ^[1454] Code Editor Key Bindings ^[1400] Code Editor Context Menu ^[1404]

11.2.3.1 Syntax Highlighting

The Code Editor highlights - in colored text - the standard code syntax of most language file formats supported by Enterprise Architect. You can define how the Code Editor implements syntax highlighting, through the Code Editors page of the Options dialog.

Topics:

Topic	Detail	See also
Supported Languages	<ul style="list-style-type: none"> • Ada (.ada, .ads, .adb) • ActionScript (.as) • BPEL Document (.bpel) • C++ (.h, .hh, .hpp, .c, .cpp, .cxx) • C# (.cs) • Delphi/Pascal (.pas) • Diff/Patch Files (.diff, .patch) • Document Type Definition (.dtd) • DOS Batch Files (.bat) • DOS Command Scripts (.cmd) • HTML (.html) • Interface Definition Language (.idl, .odl) • Java (.java) • Javascript (.javascript) • JScript (.js) • Modified Backus-Naur Form Grammar (.mbnf) • PHP (.php, .php4, .inc) • Python (.py) • Standard Generalized Markup Language (.sgml) • Structured Query Language (.sql) • SystemC (.sc) • Visual Basic 6 (.bas) • VB.NET (.vb) • VBScript (.vbs) • Verilog (.v) • VHSIC Hardware Description Language (.vhdl) • Visual Studio Resource Configuration (.rc) • eXtensible Markup Language (.xml) 	Editor Language Properties <small>1520</small>

11.2.3.2 Bookmarks

Topics:

Topic	Detail	See also
Usage	<p>Bookmarks denote a line of interest in the document; you can toggle them on and off for a particular line by pressing (Ctrl+F2)</p> <p>Additionally, you can press (F2) and (Shift+F2) to navigate to the next or previous bookmark in the document</p>	

11.2.3.3 Cursor History

Topics:

Topic	Detail	See also
Usage	<p>The Code Editor Control keeps a history of the previous 50 cursor positions</p> <p>An entry in the history list is created when:</p>	

Topic	Detail	See also
	<ul style="list-style-type: none"> The cursor is moved more than 10 lines from its previous position The cursor is moved in a <i>find/replace</i> operation <p>You can navigate to an earlier point in the cursor history by pressing (Ctrl+-), and to a later point by pressing (Ctrl+Shift+-)</p>	

11.2.3.4 Brace Matching

Topics:

Topic	Detail	See also
Usage	<p>When you place the cursor over a brace or bracket, the Code Editor highlights its corresponding partner. You can then navigate to the matching brace by pressing (Ctrl+E)</p> <pre> 27 function GetPublishedPropertyTest4: Extended; 28 function ProtectedFunctionTest: boolean; 29 procedure ProtectedProcedureTest(a: WideString); 30 function SetPublishedPropertyTest4: Extended; </pre>	

11.2.3.5 Automatic Indentation

Topics:

Topic	Detail	See also
Usage	<p>For each supported language, the Code Editor adjusts the indentation of a new line according to the presence of control statements or scope block tokens in the lines leading up to the cursor position</p> <p>For more information on customizing automatic indentation for a language, see the <i>Configuration Guide</i></p>	Code Editors <small>[1394]</small>

11.2.3.6 Commenting Selections

Topics:

Topic	Detail	See also
Usage	<p>For languages that support comments, the Code Editor can comment entire selections of code</p> <p>The Code Editor recognizes two types of commenting:</p> <ul style="list-style-type: none"> Line Commenting - entire lines are commented from the start (for example: <pre>// This is a comment</pre> Stream Commenting - sections of a line are commented from a specified start point to a specified end point (for example: <pre>/* This is a comment */</pre> 	Code Editors <small>[1394]</small>

Topic	Detail	See also
	<p>You can toggle comments on the current line or selection by pressing:</p> <ul style="list-style-type: none"> • (Ctrl+Shift+C) for line comments, or • (Ctrl+Shift+X) for stream comments <p>For more information on customizing selection commenting for each language, see the <i>Configuration Guide</i></p>	

11.2.3.7 Scope Guides

Topics:

Topic	Detail	See also
Usage	<p>If the cursor is placed over an indentation marker, the Code Editor performs a 'look back' to find the line that started the scope at that indentation level; if the line is found and is currently on screen, it is highlighted in light blue</p> <pre> 93 // If there were any answers, then return a packet, if not then just return null 94 // to indicate the server has no response 95 if (answers.size() > 0) 96 { 97 DNSPacket responsePacket = Helpers.createResponsePacket(answers, this.theS 98 responsePacket.queryID = receivedPacket.queryID; 99 100 return responsePacket; 101 } 102 else 103 { 104 return null; 105 } </pre> <p>Alternatively if the line is off screen, a calltip is displayed advising of the line number and contents.</p> <pre> 93 // If there were any answers, then return a packet, if not then just return null 94 // to indicate the server has no response 95 if (answers.size() > 0) 96 { 97 DNSPacket responsePacket = Helpers.createResponsePacket(answers, this.theS 98 responsePacket.queryID = receivedPacket.queryID; 99 100 return responsePacket; 101 } 102 else 103 { 104 return null; 105 } </pre> <p>Line 97: private DNSPacket processQuery(DNSPacket receivedPacket)</p>	

11.2.3.8 Zooming

Topics:

Topic	Detail	See also
Usage	<p>You can zoom into and out of the contents of the Code Editor using:</p> <ul style="list-style-type: none"> • (Ctrl) + keypad (+) and • (Ctrl) + keypad (-) 	

Topic	Detail	See also
	Zoom can be restored to 100% using (Ctrl) + keypad (/)	

11.2.3.9 Line Selection

Topics:

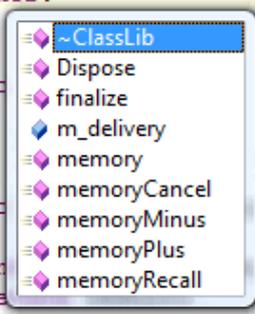
Topic	Detail	See also
Usage	If you want to move the cursor to a specific line of code, press (Ctrl + G) and, in response to the prompt, type in the line number Press the OK button; the editor displays the specified line of code with the cursor at the left	

11.2.3.10 Intellisense

Intellisense is a feature that provides choices of code items and values as you type. Not all code editors use intellisense; those that do are indicated by an asterisk in the list of Enterprise Architect code editors. Intellisense provides you with context-based assistance through autocompletion lists, calltips and mouseover information.

Intellisense is also disabled while you record a macro in the Source Code Viewer.

Topics:

Topic	Detail	See also
Autocompleti on List	<p>An autocompletion list provides a list of possible completions for the current text; the list is automatically invoked when you enter an accessor token (such as a period or pointer accessor) after an object or type that contains members</p> <pre> 57 public void memoryRecall() 58 { 59 this. 60 } 61 62 public 63 64 } 65 66 public memoryCancel (int number1, int number2) 67 { 68 in 69 re 70 } 71 </pre>  <p>You can also invoke the autocompletion list manually by pressing (Ctrl + Spacebar); the Code Editor then searches for matches for the word leading up to the invocation point</p> <p>Select an item from the list and press (Enter) or (Tab) to insert the item into</p>	

Topic	Detail	See also
	the code; to dismiss the autocompletion list, press (Escape)	
Calltips	<p>Calltips display the current method's signature when you type the parameter list token (for example, opening parenthesis); if the method is overloaded, the calltip displays arrows that you can use to navigate through the different method signatures</p> <pre> 20 //PostDraw Adornments 21 //Stereotyped Static Adornments 22 //Add Stakeholder's STAKE 23 setpenwidth(24 // Add a the 25 startpath(26 moveto(25,37); 27 lineto(25,52); 28 endpath(); 29 strokepath(); 30 //Add tip </pre>	
Mouseover Information	<p>You can display supporting documentation for code elements (for example, attributes and methods) by hovering the cursor over the element in question</p> <pre> 11 dockable = "none"; 12 string 13 / Dock elements together. Tagged V 14 Valid Values: none, standard 15 //PreDraw Derived Attribute I </pre>	

Learn More:

- [Code Editors](#)^[1394]
- [Source Code Viewer](#)^[1408]

11.2.3.11 Code Editor Key Bindings

Key	Description
Ctrl + G	Move cursor to a specified line
↓	Move cursor down one line
Shift + ↓	Extend selection down one line
Ctrl + ↓	Scroll down one line
Alt + Shift + ↓	Extend rectangular selection down one line
↑	Move cursor up one line
Shift + ↑	Extend selection up one line
Ctrl + ↑	Scroll up one line

Key	Description
Alt + Shift + ↑	Extend rectangular selection up one line
Ctrl+ (Move cursor up one paragraph
Ctrl+Shift+ (Extend selection up one paragraph
Ctrl+)	Move cursor down one paragraph
Ctrl+Shift+)	Extend selection down one paragraph
←	Move cursor left one character
Shift + ←	Extend selection left one character
Ctrl + ←	Move cursor left one word
Ctrl + Shift + ←	Extend selection left one word
Alt + Shift + ←	Extend rectangular selection left one character
→	Move cursor right one character
Shift + →	Extend selection right one character
Ctrl + →	Move cursor right one word
Ctrl + Shift + →	Extend selection right one word
Alt + Shift + →	Extend rectangular selection right one character
Ctrl + /	Move cursor left one word part
Ctrl + Shift + /	Extend selection left one word part
Ctrl + \	Move cursor right one word part
Ctrl + Shift + \	Extend selection right one word part
Home	Move cursor to the start of the current line
Shift + Home	Extend selection to the start of the current line
Ctrl + Home	Move cursor to the start of the document
Ctrl + Shift + Home	Extend selection to the start of the document
Alt + Home	Move cursor to the absolute start of the line
Alt + Shift + Home	Extend rectangular selection to the start of the line
End	Move cursor to the end of the current line

Key	Description
Shift + End	Extend selection to the end of the current line
Ctrl + End	Move cursor to the end of the document
Ctrl + Shift + End	Extend selection to the end of the document
Alt + End	Move cursor to the absolute end of the line
Alt + Shift + End	Extend rectangular selection to the end of the line
Page Up	Move cursor up a page
Shift + Page Up	Extend selection up a page
Alt + Shift + Page Up	Extend rectangular selection up a page
Page Down	Move cursor down a page
Shift + Page Down	Extend selection down a page
Alt + Shift + Page Down	Extend rectangular selection down a page
Delete	Delete character to the right of the cursor
Shift + Delete	Cut selection
Ctrl + Delete	Delete word to the right of the cursor
Ctrl + Shift + Delete	Delete until the end of the line
Insert	Toggle overwrite
Shift + Insert	Paste
Ctrl + Insert	Copy selection
Backspace	Delete character to the left of the cursor
Shift + Backspace	Delete character to the left of the cursor
Ctrl + Backspace	Delete word to the left of the cursor
Ctrl + Shift + Backspace	Delete from the start of the line to the cursor
Alt + Backspace	Undo delete
Tab	Indent cursor one tab
Ctrl + Shift + I	Indent cursor one tab
Shift + Tab	Unindent cursor one tab

Key	Description
Ctrl + keypad(+)	Zoom in
Ctrl + keypad(-)	Zoom out
Ctrl + keypad(/)	Restore Zoom
Ctrl + Z	Undo
Ctrl + Y	Redo
Ctrl + X	Cut selection
Ctrl + C	Copy selection
Ctrl + V	Paste
Ctrl + L	Cut line
Ctrl + Shift + L	Delete line
Ctrl + T	Transpose line
Ctrl + Shift + T	Copy line
Ctrl + A	Select entire document
Ctrl + D	Duplicate selection
Ctrl + U	Convert selection to lowercase
Ctrl + Shift + U	Convert selection to uppercase
Ctrl + E	Move cursor to matching brace
Ctrl + Shift + E	Extend selection to matching brace
Ctrl + Shift + C	Toggle line comment on selection
Ctrl + Shift + X	Toggle stream comment on selection
Ctrl + F2	Toggle bookmark
F2	Go to next bookmark
Shift + F2	Go to previous bookmark
Ctrl + Shift + W	Toggle whitespace characters

Key	Description
Ctrl + Shift + L	Toggle EOL characters
Ctrl + Spacebar	Invoke autocomplete
Ctrl + -	Go backwards in cursor history
Ctrl + Shift + -	Go forwards in cursor history
F12	Start/Cancel search for keyword in file(s).
Ctrl + F	Find text
Ctrl + R	Replace

Notes:

- In addition to the following keys, you can assign (**Ctrl + Alt + n**) key combinations to macros that you define within the Source Code Editor

Learn More:

- [Editor Language Properties](#)^[1520]
- [Source Code Viewer Toolbar](#)^[1410]

11.2.3.12 Code Editor Context Menu

When working on a file with a code editor, you can access a number of options through the context menu. Right-click on a text string to display the menu.

Topics:

Topic	Detail	See also
Options	<p>The options on the menu can vary depending on which editor you are using, but should include some or all of the following:</p> <ul style="list-style-type: none"> Go to Declaration locates the declaration of a symbol in the source code Go to Definition locates the definition of a symbol in the source code (applicable to languages where symbols are declared and defined in separate places, such as C++ or Delphi) Search for '<string>' - displays a submenu that enables you to locate the search string in a range of locations: <ul style="list-style-type: none"> Find in Project Browser finds the object containing the selected text in the Project Browser Search in Open Files opens the Execution Analyzer File Search facility, then searches for the selected text string in other code files of the same type and in the same folder as the current file that are open, 	Code Editor ^[1394] Find In Files ^[1454] Pre-Defined Searches ^[479] Model Search ^[475] View the Local Variables ^[1442] Create Use Case For Method ^[1406] Marker Types ^[1642]

Topic	Detail	See also
	<p>displaying the results in Tree View; you can change the folder path, search text and file type as required within the File Search window</p> <ul style="list-style-type: none"> • Search in Files performs the same search as Search in Open Files, except that the search is in all comparable files whether they are open or not • Search in Model performs an <i>Element Name</i> search in the Model Search facility, and displays the results on the Model Search tab • EA User Guide displays the description of the code item in the <i>Enterprise Architect User Guide</i> • Google displays the results of a search on the text from a Google search • MSDN displays the results of a search on the text in the Microsoft Developer Network (MSDN) • Sun Java SE displays the results of a search on the text in the Sun Microsystems 'Sun Search' facility • Wikipedia displays any entry on the object on the <i>Wikipedia</i> web site • Koders displays the results of the search for the text string on <i>Koders.com</i> • Set Debugger to Line - (if the debugger is executing and has reached a breakpoint) moves the execution point to the current line; ensure that you do not skip over any code or declarations that affect the next section of code being debugged • Display Variable - (if the debugger is executing) opens the Locals window and highlights the local variable for the current point in the code • Create Use Case for '<string>' - displays the Create Use Case For Method dialog, through which you create a Use Case for the method containing the text string • Breakpoint - displays a submenu of options for creating a breakpoint or recording marker on the selected line of code • Open (Close) IME - opens the Input Method Editor to enable you to enter text in your selected foreign language script (such as Japanese); you set the keyboard language using the Windows Control Panel - Regional and Language Options facility • Line Numbers - (Script Editor only) shows or hides the code line numbers on the left hand side of the editor screen <p>The remaining six context menu options provide simple editing functions</p>	
Script Editor	<p>If you select the Search for '<strings>' context menu option while working in the Script Editor, a slightly different submenu displays</p> <p>The Search in Scripts option opens the Execution Analyzer File Search facility, setting the Search Path field to Search in Scripts and the Search Text field to the selected text, then searching all scripts for the text string and displaying the results of the search in Tree View</p>	Script Editor ^[1407]

Notes:

- The options in the lower half of the menu (after **Search in Model**) are configurable; you can add new search tools or remove existing ones by editing the *searchProviders.xml* file in the *Sparx Systems > EA > Config* folder - this file is in OpenSearch description document format

Learn More:

- [OpenSearch Description Document Format](#) (Online Resource)

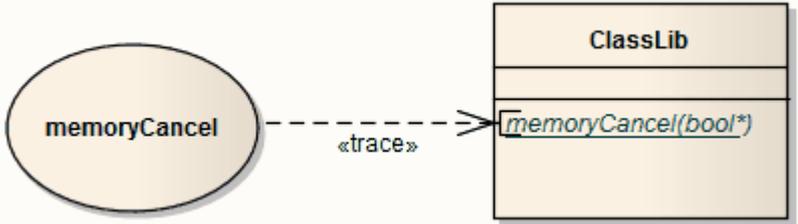
11.2.3.12.1 Create Use Case for Method

The code editor context menu enables you to create a Use Case element for a method that you select from the code. You can also link the Use Case directly to the method in a diagram, and hide any attributes or methods that are not the targets of feature links.

How To:

To create a Use Case for a method, follow the steps below:

Step	Action	See also
1	If you want to depict the Use Case and its link to the method in a diagram, click on the diagram name in the Project Browser	
2	In the code editor, right click on either the method name or any part of the method body, and select the Create Method for <methodname> context menu option The Create Use Case for Method dialog displays	
3	The basic function of this dialog is to create a Use Case for the selected method: <ul style="list-style-type: none"> If this is all that is required, click on the OK button. The Use Case element is created in the Project Browser, in the same package as the parent Class for the method, and with the same name as the method If you intend to make the relationship tangible, continue with the procedure 	
4	To create a Trace connector linking the Use Case to the method, select the Link Use Case to Method check box.	
5	To add the method's parent Class to the diagram, if it is not already there, select the Add Class to Diagram check box.	
6	To add the newly-created Use Case to the diagram, select the Add Use Case to Diagram check box. This would now show the Use Case, Class and Trace connector on the diagram.	
7	To only show the features (attributes and methods) of the parent Class that are the targets of 'link to feature' relationships, select the Display only linked features in Class checkbox. The Class might contain any number of attributes and methods, but those without a 'link to feature' relationship are hidden.	Connect to Element Feature ^[742]
8	Click on the OK button to create and depict the Use Case and relationship. If you selected all options, the diagram now contains linked elements	

Step	Action	See also
	<p>resembling the following:</p> 	

Learn More:

- [Code Editor Context Menu](#)^[1404]

11.2.3.13 Script Editor

The *Script Editor* enables you to edit scripts, and to run and stop an open script.

The editor is based on, and provides the facilities of, the common **Code Editor**.

To open the Script Editor, double-click on the required script in the **Scripts** tab of the **Scripting window**. The editor opens in the main work area.

Use to:

- Save changes to the current script
- Save the current script under a different name
- Run the script
- Stop the executing script
- View the script output in the **Scripts** tab of the **Output** window

Topics:

Topic	Detail	See also
Enterprise Architect Script Objects	<p>Enterprise Architect adds to the available functionality and features of the editor script language by providing inbuilt objects. These are either <i>Type Libraries</i>, providing intellisense for editing purposes, or <i>Runtime objects</i>, providing access to objects of the types described in the Type Libraries.</p> <p>The available intellisense scripting objects are:</p> <ul style="list-style-type: none"> • EA • MathLib • System • The runtime scripting objects (below) <p>The available runtime scripting objects are:</p> <ul style="list-style-type: none"> • Repository (Type: IDualRepository) - this is the Enterprise Architect automation interface • Maths (Type: IMath) • Session (Type: ISession) 	Automation Interface ^[1822]

Topic	Detail	See also
Script Editing Intellisense (Required Syntax)	<p>Intellisense is available not only in the Script Editor, but also in the Script Console. Intellisense at its most basic is presented for the inbuilt functionality of the script engine. For intellisense on the additional Enterprise Architect scripting objects listed above, you must declare variables according to syntax that specifies a type. It is not necessary to use this syntax to execute a script properly. It is only present so that the correct intellisense can be displayed for an item. The syntax can be seen in the above diagram in, for example:</p> <pre>Dim e as EA.Element</pre> <p>Then, when you type, in this case, e. the editor displays a list of member functions and properties of e's type.</p> <p>You select one of these to complete the line of script. You might, therefore, type:</p> <pre>VTrace(e.</pre> <p>As you type the period, the editor presents the appropriate list and you might double-click on, for example, Abstract. This is inserted in the line, and you continue to type or select the rest of the statement. In this case, adding the end space and parenthesis.</p> <pre>VTrace(e. Abstract)</pre>	
Keystrokes	<p>In the Script Editor or Console, intellisense is presented on the following keystrokes.</p> <ul style="list-style-type: none"> • Press (.) (period) after an item to list any members for that item's type • Press (Ctrl) +(Spacebar) on a word to list any intellisense items with a name starting with the string at the point keystroke was pressed • Press (Ctrl) +(Spacebar) when not on a word to display any available top level intellisense items - these are the intellisense objects described above plus any built-in methods and properties of the current scripting language 	

Notes:

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- Enterprise Architect's scripting supports declaring variables to match the EA types. This enables the editor to present **intellisense**, but is not necessary for executing the script

Learn More:

- [Code Editor](#) 
- [Script Tabs](#) 

11.2.3.14 The Source Code Viewer

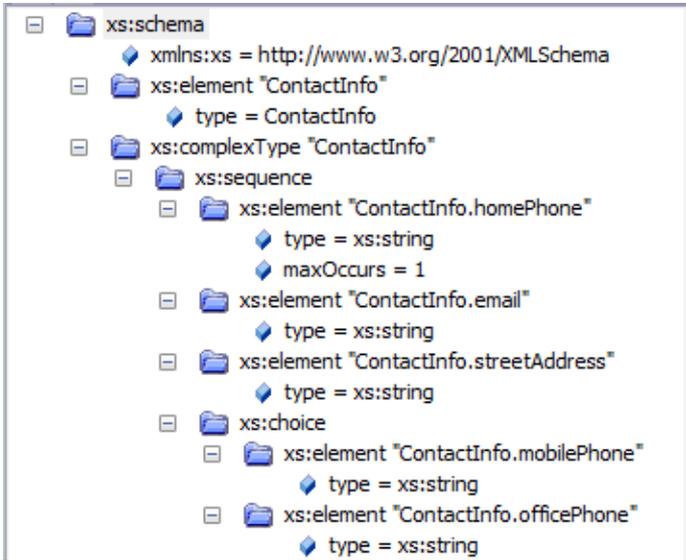
Access: **View | More Element Tools | Source Code** (Alt+7)
Tools | Open Source File (Ctrl+Alt+O)

Use the Source Code viewer to view any source code files for an element. If a Class is selected, the Source Code viewer shows the source code for that Class, provided it has already been *generated*. For C++ a second tab displays to show the implementation file.

The Source Code viewer also displays any DDL generated for a selected table in your diagram.

A number of options change the way the Source Code viewer works. They can be altered via the Options dialog (select the **Tools | Options | Source Code Engineering | Code Editors** menu option). There are also many options for developing the code, available through the code editor context menu and through the Toolbar.

Topics:

Topic	Detail	See also
Source Code Viewer	<p>By default the Source Code viewer is set to:</p> <ul style="list-style-type: none"> • Parse all opened files, and show a tree of the results • Show line numbers <p>If you are editing an XML file, the structure tree is presented in a folder hierarchy rather than a Class structure hierarchy, as follows:</p>  <pre> xs:schema ├── xmlns:xs = http://www.w3.org/2001/XMLSchema ├── xs:element "ContactInfo" │ └── type = ContactInfo ├── xs:complexType "ContactInfo" │ └── xs:sequence │ ├── xs:element "ContactInfo.homePhone" │ │ ├── type = xs:string │ │ └── maxOccurs = 1 │ ├── xs:element "ContactInfo.email" │ │ └── type = xs:string │ ├── xs:element "ContactInfo.streetAddress" │ │ └── type = xs:string │ └── xs:choice │ ├── xs:element "ContactInfo.mobilePhone" │ │ └── type = xs:string │ └── xs:element "ContactInfo.officePhone" │ └── type = xs:string </pre>	
External File Editor	<p>If you intend to edit external code, XML and DDL files (that is, code not selected from the Project Browser) use the Tools Open Source File menu option, or press (Ctrl+Alt+O).</p> <p>If you try to close the window or select another file, the editor prompts you to save your changes.</p>	
File Parsing	<p>The Source Code viewer parses files for a number of reasons. The first is to enable it to jump to the location in the file at which the currently selected item is found (but not when editing external files).</p> <p>Additionally, parsing displays a structure tree showing an overview of the file in a similar fashion to the main Project Browser. You can also select anything in that and jump to the appropriate line in the editor.</p> <p>The viewer cannot parse DDL, and therefore does not show the structure tree for a DDL file.</p>	

Notes:

- You view source code for an element by selecting menu options in a number of places, or by pressing

either (**Ctrl+E**) or (**F12**) . If the element does not have a generation file (that is, code has not been or cannot be generated, such as for a Use Case), Enterprise Architect checks whether the element has a link to either an operation or an attribute of another element. If such a link exists, and that other element has source code, the code for that element displays

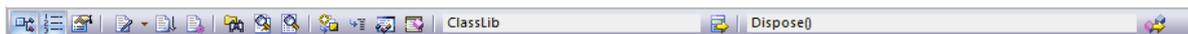
Learn More:

- [Connect to Element Feature](#)^[742]
- [Generate Source Code](#)^[1490]
- [Code Editors](#)^[1519]
- [Code Editor Context Menu](#)^[1404]
- [Source Code Viewer Toolbar](#)^[1410]

11.2.3.14.1 Source Code Viewer Toolbar

The toolbar buttons in the **Source Code** viewer enable you to edit, view and interact with the code contained in the **Source Code** viewer. The function of each button, from left to right, is described below:

The Toolbar:



Topics:

Topic	Detail	See also
Toolbar Options	<ul style="list-style-type: none"> • Structure Tree - shows or hides the element hierarchy panel (the left panel of the Source Code viewer) • Line Numbers - shows or hides the line numbers against the lines of code • Source Code Engineering Properties - displays the Source Code Engineering page of the Options dialog, from which you can configure display and behavior options for source code engineering • Editor Functions - provides quick access to the following functions: <ul style="list-style-type: none"> • Open Corresponding File - opens the header or implementation file associated with the currently-open file • Go to Matching Brace - for a selected opening or closing brace, highlights the corresponding closing or opening brace in the pair • Go to Line - displays a small dialog on which you select the number of the line to highlight; click on the OK button to move the cursor to that line • Cursor History Previous - the Source Code viewer keeps a history of the previous 50 cursor positions, creating a record when the cursor is moved either more than 10 lines away from its previous position, or in a find-and-replace operation; the menu option moves the cursor to the position in the immediately-previous cursor history record • Cursor History Next - moves the cursor to the position in the immediately-following cursor history record • Record Macro - record the subsequent keystrokes to be saved as a macro • Stop Recording and Save Macro - stop recording the keystrokes and specify a name for the macro, on the Save 	Source Code Engineering ^[1516] Code Templates Editor ^[1485] Find in Files ^[1454] Model Search ^[475]

Topic	Detail	See also
	<p>Macro dialog</p> <ul style="list-style-type: none"> • Play Macro - execute the macro to repeat the saved keystrokes, if necessary selecting the macro from the Open Macro dialog • Toggle Line Comment - comments out (//) or re-establishes the code for each full line in which text is highlighted • Toggle Stream Comment - inserts a stream comment (/* */) at the cursor position or comments out the highlighted characters and lines, or re-establishes the commented text as code • Toggle Whitespace Characters - shows or hides the spacing characters: --> (tab space) and . (character space) • Toggle EOL Characters - shows or hides the end-of-line characters: CR (carriage return) and LF (line feed) • Save Source and Resynchronize Class - saves the source code and resynchronizes the Class • Code Templates - accesses the Code Templates Editor • Find in Project Browser - for a selected line of code, highlights the appropriate structure in the Project Browser; if there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the appropriate structure from which you can select the required one • Search in Files - searches for the selected object name in associated files and displays the results of the search on the File Search window • Search in Model - searches for the selected text throughout the model, and displays the results of the search on the Model Search window. • Go to Declaration - locates the declaration of a symbol in the source code • Go to Definition - locates the definition of a symbol in the source code (applicable to languages where symbols are declared and defined in separate files e.g. C++, Delphi) • Autocomplete List - displays the autocompletion list of possible values; double-click on a value to select it • Parameter Information - when the cursor is between the parentheses of an operation's parameter list, displays the operation's signature, highlighting the current parameter • Find Current Class in Project Browser - displays the name of the currently-selected Class in the code, and highlights that name in the Project Browser; if there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the Class from which you can select the required one • Find Member - displays the name of the currently-selected attribute or method in the code, and highlights that name in the Project Browser; if there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the feature from which you can select the required one 	

Notes:

- The Record Macro function disables **Intellisense** while the macro is being recorded
- You can **assign key strokes** to execute the macro, instead of using the toolbar drop-down and Open

Macro dialog

Learn More:

- [Intellisense](#)^[1399]
- [Editor Language Properties](#)^[1520]

11.2.3.15 Find and Replace

Each of Enterprise Architect's code editors facilitates searching for and replacing terms in the editor, through the **Find and Replace** dialog.

Access:

Highlight the required text string and press:

- **(Ctrl+F)** for the find controls only, or
- **(Ctrl+R)** for both find and replace controls

In each instance, the **Find what** field is populated with the text currently selected in the editor. If no text is selected in the editor, the **Find what** field is populated with the word at the current cursor position. If no word exists at the current cursor position, the last searched-for term is used.

Basic Operations

Commands:

Command	Action	See also
Find Next	Locate and highlight the next instance (relative to the current cursor position) of the text specified in the Find what field.	
Replace	Replace the current instance of the text specified in the Find what field with the text specified in the Replace with field, and then locate and highlight the next instance (relative to the current cursor position) of the text specified in the Find what field.	
Replace All	Automatically replace all instances of the text specified in the Find what field with the text specified in the Replace with field.	

Options:

Option	Action	See also
Match Case	Specify that the case of each character in the text string in the Find what field is significant when searching for matches in the code.	
Match whole word	Specify that the text string in the Find what field is a complete word and should not be matched with instances of the text that form part of a longer string. For example, searches for ARE should not match those letters in instances of the words AREA or ARENA.	
Search up	Perform the search from the current cursor position up to the start of the file, rather than in the default direction of current	

Option	Action	See also
	cursor position to end of file.	
Use Regular Expressions	Evaluate specific character sequences in the Find what and Replace with fields as <i>Regular Expressions</i> (see below).	

Topics:

Topic	Detail	See also												
Regular Expressions	<p>A Regular Expression is a formal definition of a search pattern, which can be used to match specific characters, words or patterns of characters. For the sake of simplicity, the Code Editor's 'find and replace' mechanism supports only a subset of the standard Regular Expression grammar.</p> <p>Text in the Find what and Replace with fields is only interpreted as a Regular Expression if the Use Regular Expressions checkbox is selected in the Find and Replace dialog.</p>													
Meta sequences	<p>If the Use Regular Expressions checkbox is selected, most characters in the Find what field are treated as literals (that is, they match only themselves). The exceptions are called <i>metasequences</i>. Each <i>metasequence</i> recognized in the Code Editor Find and Replace dialog is described in the following table.</p> <table border="1"> <thead> <tr> <th>Metasequence</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>.</td> <td>Represents any single character. For example: .at is matched to cat, hat, bat and rat.</td> </tr> <tr> <td>\<</td> <td>Indicates that the text is the start of a word. For example: \<cat is matched to catastrophe and cataclysm, but not concatenate.</td> </tr> <tr> <td>\></td> <td>Indicates that the text is the end of a word. For example: hat\> is matched to that and chat, but not hate.</td> </tr> <tr> <td>(...)</td> <td>Indicates alternative single characters that can be matched. The characters can be specific - (chr) - or in an alphabetical or numerical range - (a-m) . For example: (hc) at is matched to hat and cat but not bat. For example: (a-m) Class is matched to any name in the range aClass - mClass.</td> </tr> <tr> <td>(^...)</td> <td>Indicates alternative single characters that should be excluded from a match. The characters can be specific - (^chr) - or in an</td> </tr> </tbody> </table>	Metasequence	Description	.	Represents any single character. For example: .at is matched to cat , hat , bat and rat .	\<	Indicates that the text is the start of a word. For example: \<cat is matched to catastrophe and cataclysm , but not concatenate .	\>	Indicates that the text is the end of a word. For example: hat\> is matched to that and chat , but not hate .	(...)	Indicates alternative single characters that can be matched. The characters can be specific - (chr) - or in an alphabetical or numerical range - (a-m) . For example: (hc) at is matched to hat and cat but not bat . For example: (a-m) Class is matched to any name in the range aClass - mClass .	(^...)	Indicates alternative single characters that should be excluded from a match. The characters can be specific - (^chr) - or in an	
Metasequence	Description													
.	Represents any single character. For example: .at is matched to cat , hat , bat and rat .													
\<	Indicates that the text is the start of a word. For example: \<cat is matched to catastrophe and cataclysm , but not concatenate .													
\>	Indicates that the text is the end of a word. For example: hat\> is matched to that and chat , but not hate .													
(...)	Indicates alternative single characters that can be matched. The characters can be specific - (chr) - or in an alphabetical or numerical range - (a-m) . For example: (hc) at is matched to hat and cat but not bat . For example: (a-m) Class is matched to any name in the range aClass - mClass .													
(^...)	Indicates alternative single characters that should be excluded from a match. The characters can be specific - (^chr) - or in an													

Topic	Detail	See also												
	<table border="1" data-bbox="496 315 1179 1211"> <thead> <tr> <th data-bbox="496 315 667 398">Metasequence</th> <th data-bbox="667 315 1179 398">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="496 398 667 633"></td> <td data-bbox="667 398 1179 633"> alphabetical or numerical range - (^a-m) . For example: (^hc) at is matched to rat and bat but hat and cat are excluded. For example: (^a-m) Class is matched to any name in the range nClass to zClass but aClass to mClass are excluded. </td> </tr> <tr> <td data-bbox="496 633 667 689">^</td> <td data-bbox="667 633 1179 689">Matches the start of a line.</td> </tr> <tr> <td data-bbox="496 689 667 745">\$</td> <td data-bbox="667 689 1179 745">Matches the end of a line.</td> </tr> <tr> <td data-bbox="496 745 667 981">*</td> <td data-bbox="667 745 1179 981"> Matches the preceding character (or character set) 0 or more times. For example: ba*t is matched to bt, bat, baat, baaat and so on. For example: b(ea) *t is matched to bt, bet, bat, beat, beet, baat and so on. </td> </tr> <tr> <td data-bbox="496 981 667 1211">+</td> <td data-bbox="667 981 1179 1211"> Matches the preceding character (or character set) 1 or more times. For example: ba+t is matched to bat, baat and baaat but not bt. For example: b(ea) +t is matched to bet, bat, beat, beet and baat but not bt. </td> </tr> </tbody> </table> <p data-bbox="496 1234 1179 1350">If a single character metasequence is preceded with a backslash (\) it is treated as a literal character. For example: c(at\) matches c(at) as the square brackets are treated literally.</p> <p data-bbox="496 1368 1179 1554">When the Use Regular Expressions checkbox is selected, a metasequence helper menu is available to the right of both of the Find what and Replace with fields. Selecting a metasequence from this menu inserts the metasequence into the field, replacing or wrapping the currently selected text as appropriate.</p>	Metasequence	Description		alphabetical or numerical range - (^a-m) . For example: (^hc) at is matched to rat and bat but hat and cat are excluded. For example: (^a-m) Class is matched to any name in the range nClass to zClass but aClass to mClass are excluded.	^	Matches the start of a line.	\$	Matches the end of a line.	*	Matches the preceding character (or character set) 0 or more times. For example: ba*t is matched to bt , bat , baat , baaat and so on. For example: b(ea) *t is matched to bt , bet , bat , beat , beet , baat and so on.	+	Matches the preceding character (or character set) 1 or more times. For example: ba+t is matched to bat , baat and baaat but not bt . For example: b(ea) +t is matched to bet , bat , beat , beet and baat but not bt .	
Metasequence	Description													
	alphabetical or numerical range - (^a-m) . For example: (^hc) at is matched to rat and bat but hat and cat are excluded. For example: (^a-m) Class is matched to any name in the range nClass to zClass but aClass to mClass are excluded.													
^	Matches the start of a line.													
\$	Matches the end of a line.													
*	Matches the preceding character (or character set) 0 or more times. For example: ba*t is matched to bt , bat , baat , baaat and so on. For example: b(ea) *t is matched to bt , bet , bat , beat , beet , baat and so on.													
+	Matches the preceding character (or character set) 1 or more times. For example: ba+t is matched to bat , baat and baaat but not bt . For example: b(ea) +t is matched to bet , bat , beat , beet and baat but not bt .													
Tagged Regions	<p data-bbox="496 1570 1179 1778">When 'find and replacing' with Regular Expressions, up to nine sections of the original term can be substituted into the replacement term. The metasequences \(and \) denote the start and the end of a <i>tagged region</i>. The section of the matched text that falls within the tagged region can be included in the replacement text with the metasequence \n (where <i>n</i> is the tagged region number between 1 and 9).</p> <p data-bbox="496 1796 639 1825">For example:</p> <p data-bbox="496 1843 1179 1899">Find what: \ ((A- Za- z) +\) ' s t h i n g s</p> <p data-bbox="496 1917 1179 1951">Replace with: i t e m s t h a t b e l o n g t o \ 1</p> <p data-bbox="496 1968 1179 2002">Original text: These are all Michael's things.</p> <p data-bbox="496 1998 1179 2031">Replaced text: These are all items that belong to</p>													

Topic	Detail	See also
	Michael.	

11.2.4 Build

The topics in this section describe how you specify the commands to build the project or package.

- [Add Comands](#) ^[1415]
- [Recursive Builds](#) ^[1416]

11.2.4.1 Add Commands

Topics:

Topic	Detail	See also
Usage	<p>The Build page enables you to enter commands to build your project</p> <p>You can run a single program by selecting Process or multiple commands by selecting Batch File</p> <p>You can use Enterprise Architect Local Paths and environment variables in composing your command line(s)</p>	
Execute Command As:	<p>Batch File</p> <p>Use this option to enter multiple commands, which are executed as entered in a command window; the command window has access to your environment variables</p> <p>Process</p> <p>Use this option to run a single program</p> <p>The command should specify the path to the program, plus any command line arguments; if the executable path or any arguments contain spaces, they must be surrounded by quotes</p>	
Output Parser	<p>This enables you to select a method for automatically parsing the compiler output</p> <p>If you select this option, output from the script is logged in the System Output window; Enterprise Architect parses the output according to the syntax you specify</p> <p>To view the output, select the View System Output menu option</p> <p>If you double-click on an error line, Enterprise Architect loads the appropriate source file and positions the cursor on the line at which the error is reported</p>	The Output Window ^[127]

Learn More:

- [Local Paths](#) ^[1523]

11.2.4.2 Recursive Builds

Topics:

Topic	Detail	See also
Usage	<p>For any project you can apply the command entered in the build script to all sub folders of the initial directory by specifying the token <code>%r</code> immediately preceding the files to be built</p> <p>The effect of this is Enterprise Architect iteratively replaces the token with any subpath found under the root and executes the command again</p>	

11.2.5 Clean

The topics in this section describe how you specify the commands to clean the project prior to a build.

- [Add Clean Comands](#) ^[1416]
- [Recursive Builds](#) ^[1416]

11.2.5.1 Add Clean Commands

This topic explains how you enter a command to clean your project. When you execute the **Rebuild** menu option on a script the command you specify here is executed followed by the build command.

Topics:

Topic	Detail	See also
Usage	This is the command that is executed when you click the Execution Analyzer toolbar Clean button or select Clean from the script context menu	
Example	<code>devenv.com /Clean Debug MyProject.sln</code>	

11.2.6 Debugging

This section describes how you define the debugging actions:

- [How it works](#) ^[1416]
- [Setup for Debugging](#) ^[1417]
- [Tracepoint Output](#) ^[1435]
- [Breakpoint and Marker Management](#) ^[1436]
- [Debugging Actions](#) ^[1441]
- [Recording Actions](#) ^[1445]

11.2.6.1 How it Works

The Model Driven Development Environment provides Debuggers for the following frameworks:

- Microsoft Native Code applications
- Microsoft .NET applications
- Java applications
- PHP.exe applications

Checklist:

Step	Action	See also
1	A model must be open	
2	Ensure any source code for the areas of interest have been generated, or imported into the model	Synchronize Model and Code ^[1506]

Option 1 - No Analyzer Script:

If you have *not* configured an Analyzer Script, you can still debug a process that is running.

Step	Action	See also
1	Optionally set breakpoints before debugging	
2	Use the Attach button on the Debug toolbar <ul style="list-style-type: none"> Select the debugging platform Select the process to debug 	

Option 2 - Analyzer Script, with Debug Page Configured:

Step	Action	See also
1	Ensure the application has been built You can do this internally using the Build Script, or you can build the application externally; the important thing is that the application is built on the latest versions of the source	Build ^[1415]
2	Set breakpoints	Set Code Breakpoints ^[1435]
3	Start the debugger	Start & Stop Debugger ^[1447]

11.2.6.2 Setup for Debugging

To begin debugging you must specify

- The debugger to use
- The application path
- Runtime options, if applicable

You specify these on the Debug page of the Analyzer Script dialog.

Learn More:

- [Operating System Specific Requirements](#) ^[1417]

11.2.6.2.1 Operating System Specific Requirements

Important - Please read the information provided in this topic.

Topics:

Topic	Detail	See also
Prerequisites	Creation of an Analyzer script and configuration of the Debug command in that script	
Supported Platforms	<p>Enterprise Architect supports debugging on these platforms:</p> <p>.NET</p> <ul style="list-style-type: none"> • Microsoft™ .NET Framework 1.1 and later, including .NET 4 • Language support: C, C#, C++, J#, Visual Basic <p>Java</p> <ul style="list-style-type: none"> • Java 2 Platform Standard edition (J2SE) version 5.0 • J2EE JDK 1.4 and above • Requires previous installation of the Java Runtime Environment and Java Development Kit from Sun Microsystems™ <p>Debugging is implemented through the Java Virtual Machine Tools Interface (JVMTI), which is part of the Java Platform Debugger Architecture (JPDA); the JPDA is included in the J2SE SDK 1.3 and later</p> <p>Windows for Native Applications</p> <p>Enterprise Architect supports debugging native code (C, C++ and Visual Basic) compiled with the Microsoft™ compiler where an associated PDB file is available</p> <p>You can import native code into your model, and record the execution history for any Classes and methods</p> <p>You can also generate Sequence diagrams from the resulting execution path</p> <p>PHP</p> <p>Enterprise Architect enables you to perform local and remote debugging with PHP.exe</p>	

Notes:

- Debugging under Windows Vista (x64) - if you encounter problems debugging with Enterprise Architect on a 64-bit platform, you should build a Win32 platform configuration in Visual Studio; that is, do not specify **ANY-CPU**, specify **WIN32**

11.2.6.2.1.1 UAC-Enabled Operating Systems

The Microsoft operating systems *Windows Vista* and *Windows 7* provide User Account Control (UAC) to manage security for applications.

The Enterprise Architect Visual Execution Analyser is UAC-compliant, and users of UAC-enabled systems can perform operations with the Visual Execution Analyser and related facilities under accounts that are members of only the *Users* group.

However, when attaching to processes running as services on a UAC-enabled operating system, it might be necessary to log in as an Administrator.

How To:

To log in as an Administrator, follow the step below:

Step	Action	See also
1	Before you run Enterprise Architect, right-click on the Enterprise Architect icon on the desktop and select the Run as administrator option	

Alternatively, to edit or create a link to Enterprise Architect and configure the link to run as an Administrator, follow the steps below:

Step	Action	See also
1	Right-click on the Enterprise Architect icon and select the Properties menu option The Enterprise Architect Properties dialog displays	
2	Click on the Advanced button The Advanced Properties dialog displays	
3	Select the Run as administrator checkbox	
4	Click on the OK button, and again on the Enterprise Architect Properties dialog	

11.2.6.2.1.2 WINE Debugging

How To:

To configure Enterprise Architect to debug under WINE, follow the steps below:

Step	Action	See also
1	At the command line, run <code>\$ winecfg</code>	
2	Set the library overrides for <code>dbghelp</code> to (native, builtin) , and accept the warning about overriding this DLL	
3	Set <code>dbghelp</code> to native by using <code>winecfg</code>	
4	Copy the application source code plus executable(s) to your bottle The path must be the same as the compiled version; that is: If Windows source = <code>C:\Source\SampleApp</code> , under Crossover it must be <code>C:\Source\SampleApp</code>	
5	Copy any Side-By-Side assemblies that are used by the application	
6	Import the source code into Enterprise Architect (optional)	
7	Create a build script on a package Set the path of the application on the Debug tab, and set the Use Debugger field to Microsoft Native	Build ^[1415]
8	Open the Profiler (Analyzer Profile)	Visual Execution Profiler ^[1658]

Step	Action	See also
9	Click on the Launch button (first button on the Profiler window)	
10	If the sample didn't start, click on the Sampling button (third button on the Profiler window)	
11	Once you have finished profiling, shut down the application (<i>not</i> Enterprise Architect)	
12	View the Sampler report by clicking the View Report button (fifth button on the Profiler window)	

Topics:

Topic	Detail	See also
Access Violation Exceptions	<p>Due to the manner in which WINE handles direct drawing and access to DIB data, an additional option is provided on the Debug window toolbar drop-down menu to ignore or process access violation exceptions thrown when your program directly accesses DIB data</p> <p>Select this option to catch genuine (unexpected) access violations; deselect it to ignore expected violations</p> <p>As the debugger cannot distinguish between expected and unexpected violations, you might have to use trial and error to capture and inspect genuine program crashes</p>	

Notes:

- If WINE crashes, the back traces might not be correct
- If you are using MFC remember to copy the debug side-by-side assemblies to the *C:\windows\winsxs* directory
- To add a windows path to WINE, modify the Registry entry:

```
HKEY_LOCAL_MACHINE\System\Current Control Set\Control\Session
Manager\Environment
```

11.2.6.2.2 Microsoft C++ and Native (C, VB)**Topics:**

Topic	Detail	See also
Usage	<p>You can debug native code only if there is a corresponding PDB file for the executable; you normally create the PDB file as a result of building the application</p> <p>The build should include full debug information and there should be no optimizations set</p> <p>The script must specify two things to support debugging:</p> <ul style="list-style-type: none"> • The path to the executable • Microsoft Native as the debugging platform 	General Setup ^[142]

11.2.6.2.2.1 General Setup

This is the general setup for debugging Microsoft Native Applications (C++, C, Visual Basic). You have two options when debugging:

- Debug an application
- Attach to an application that is running

Option 1 -Debug an application

Field	Action	See also
Debugger	Select Microsoft Native as the debugging platform	
Mode	Select the Run radio button	
Default Directory	This is set as the default directory for the process being debugged	
Application	Select and enter either the full or the relative path to the application executable	
Command Line Arguments	Parameters to pass to the application at startup	
Show Console	Create a console window for the debugger; not applicable for attaching to a process	
Symbol Search Path	Specify any additional paths to locate debug symbols for the debugger; separate the paths with a semi-colon	

Option 2 - Attach to an application that is running

Field	Action	See also
Debugger	Select Microsoft .NET as the debugging platform	
Mode	Select the Attach to Process radio button	
Symbol Search Path	Specify any additional paths to locate debug symbols for the debugger You could specify a symbol server here if you prefer; separate the paths with a semi-colon or comma	

11.2.6.2.2.2 Debug Symbols

Topics:

Topic	Detail	See also
Abstract	For applications built using <i>Microsoft Platform SDK</i> , Debug Symbols are written to an application PDB file when the application is built The <i>Debugging Tools for Windows</i> , an API used by the Visual Execution Debugger, uses these symbols to present meaningful information to Execution Analyzer controls These symbols can easily get out of date and cause errant behaviour - the debugger might highlight the wrong line of code in the editor whilst at a	

Topic	Detail	See also
	<p>breakpoint; it is therefore best to ensure the application is built prior to any debugging or recording session</p> <p>The debugger must inform the API how to reconcile addresses in the image being debugged; it does this by specifying a number of paths to the API that tell it where to look for PDB files</p> <p>For system DLLs (<i>kernel32</i>, <i>mfc90ud</i>) for which no debug symbols are found, the Call Stack shows some frames with module names and addresses only</p> <p>You can supplement the symbols translated by passing additional paths to the API; you pass additional symbol paths in a semi-colon separated list in the Debug tab</p>	

11.2.6.2.3 Java

This section describes how to set up Enterprise Architect for debugging Java applications and Web Servers.

Learn More:

- [General Setup for Java](#) ^[1422]
- [Advanced Techniques](#) ^[1423]
- [Working with Java Web Servers](#) ^[1425]

11.2.6.2.3.1 General Setup for Java

This is the general setup for debugging Java Applications. You have two options when debugging.

- Debug an Application
- Attach to an application that is running

Option 1 - Debug an Application

Field	Action	See also
Debugger	Select Java	
Mode	Select Run	
Default Directory	This path is added to the class path property when the Java Virtual Machine is created	
Application Class	<p>Identify the fully qualified Class name to debug; the Class must have a method declared with the following signature:</p> <pre>public static void main(String());</pre> <p>Application Class <input type="text" value="samples.Collector"/></p> <p>Command Line Arguments: </p>	

Field	Action	See also
Java Virtual Machine Options	<p>Specify command line options for Virtual Machine creation</p> <p>You also must provide a parameter (jre) that is a path to be searched for the <code>jvm.dll</code>; this is the DLL supplied as part of the Java runtime environment or Java JDK from Sun Microsystems™</p> <p>Java Virtual Machine Options:</p> <pre>jre=%JAVA%\jre,-Djava.class.path=%classpath%;;</pre> <p>In the example above, a virtual machine is created with a new Class path property that comprises any paths named in the <code>CLASSPATH</code> environment variable plus the default directory</p> <p>If no Class path is specified, the debugger always creates the virtual machine with a Class path property equal to any path contained in the environment variable plus the path entered in the default working directory of this script</p> <p>If source files and <code>.class</code> files are located under different directory trees, the Class path property MUST include both root path(s) to the source and root path(s) to binary class files</p>	

Option 2 - Attach to Virtual Machine

There is very little to specify when attaching to a VM; however, the VM must have the Sparx Systems debugging agent loaded.

Field	Action	See also
Debugger	Select Java	Attach to Virtual Machine ^[1424]
Mode	Select Attach to Virtual Machine	Debug Tomcat Server ^[1428]

11.2.6.2.3.2 *Advanced Techniques*

In addition to the standard Java debugging techniques, you can:

- [Attach to a Virtual Machine](#)^[1424]
- [Debug Internet Browser Java Applets](#)^[1424]

Topics:

Topic	Detail	See also
Abstract	<p>You can debug a Java application by attaching to process that is hosting a Java Virtual Machine; you might want to do this for attaching to a webserver such as Tomcat or JBOSS</p> <p>The <i>Java Virtual Machine Tools Interface</i> from Sun Microsystems, is the API used by Enterprise Architect; it allows a debugging agent to be specified when the JVM is created</p> <p>To debug a running JVM from Enterprise Architect, the Sparx Systems' debugging agent must have been specified as a startup option to the JVM when it was started; how this accomplished for products such as Tomcat and JBOSS should be researched from that product's own documentation</p> <p>For <i>java.exe</i>, the command line option to load the Enterprise Architect debugging agent is:</p> <pre>- agent lib: SSJavaProfiler75</pre> <p>or:</p> <pre>- agent path: "c:\program files\sparx systems\ea\SSJavaProfiler75"</pre> <p>It is not necessary to configure an Analyzer Script when you attach to a Virtual Machine; you can just use the Attach button on one of the Analyzer toolbars</p> <p>If you configure an Analyzer Script, there are only two things that must be selected:</p> <ul style="list-style-type: none"> • Select Java as the debugging platform • Choose the Attach to Virtual Machine option 	

This topic describes the configuration requirements and procedure for debugging Java Applets running in a browser from Enterprise Architect.

How To:

To attach to the browser process hosting the Java Virtual Machine (JVM) from Enterprise Architect, follow the steps below:

Step	Action	See also
1	Ensure binaries for the applet code to be debugged have been built with debug information	
2	Configure the JVM using the Java Control Panel	
3	In the Java Applet Runtime Settings panel, click on the View button	
4	<p>On the installed version to use, include the following option in the Runtime Parameters field:</p> <pre>- agent path: c:\program files\sparx systems\ea\SSJavaProfiler75</pre>	
5	<p>In this field add the required Class paths</p> <p>At least one of these paths should include the root path of the source files to use</p>	

Step	Action	See also
	in debugging	
6	Set breakpoints	Breakpoint and Marker Management ^[1436]
7	Launch the browser	
8	Attach to the browser process from Enterprise Architect	

11.2.6.2.3.3 Working with Java Web Servers

This topic describes the configuration requirements and procedure for debugging Java web servers such as JBOSS and Apache Tomcat (both Server configuration and Windows Service configuration) in Enterprise Architect.

How To:

To attach to the process hosting the Java Virtual Machine from Enterprise Architect, follow the steps below:

Step	Action	See also
1	Ensure binaries for the web server code to be debugged have been built with debug information	
2	Launch the server with the Virtual Machine startup option described in <i>Server Configuration</i> , below	
3	Import source code into the Enterprise Architect Model, or synchronize existing code	
4	Set breakpoints	Breakpoint and Marker Management ^[1436]
5	Launch the client	
6	Attach to the process from Enterprise Architect	

Topics:

Topic	Detail	See also
Server Configuration	<p>The configuration necessary for the web servers to interact with Enterprise Architect must address the following two essential points:</p> <ul style="list-style-type: none"> Any VM to be debugged, created or hosted by the server must have the Sparx Systems Agent <code>SSJavaProfiler75</code> command line option specified in the VM startup option (that is: <code>-agentlib:SSJavaProfiler75</code>) The CLASSPATH, however it is passed to the VM, must specify the root path to the package source files 	

Topic	Detail	See also
	<p>The Enterprise Architect debugger uses the <code>java.class.path</code> property in the VM being debugged, to locate the source file corresponding to a breakpoint occurring in a Class during execution; for example, a Class to be debugged is called:</p> <p style="padding-left: 40px;">a . b . C</p> <p>This is located in physical directory:</p> <p style="padding-left: 40px;">C : \ s o u r c e \ a \ b</p> <p>So, for debugging to be successful, the CLASSPATH must contain the root path:</p> <p style="padding-left: 40px;">c : \ s o u r c e</p>	
Analyzer Script Configuration	<p>Using the Debug tab of the Build Script dialog, create a script for the code you have imported and:</p> <ul style="list-style-type: none"> • Select the Attach to process radio button and, in the field below, type attach • In the Use Debugger field, click on the drop-down arrow and select Java <p>All other fields are unimportant; the Directory field is normally used in the absence of any Class path property</p>	General Setup for Java ^[1422]

How To:

To debug, follow the steps below:

Step	Action	See also
1	<p>Ensure that the server is running, and that the server process has loaded the Sparx Systems Agent</p> <p>DLL SSJavaPr of i l e r 75. DLL</p> <p>(Use <i>Process Explorer</i> or similar tools to prove this)</p>	
2	In Enterprise Architect, open the source code and set some breakpoints	Breakpoint and Marker Management ^[1436]
3	<p>Click on the Debug button in Enterprise Architect</p> <p>The Attach To Process dialog displays</p>	Start and Stop Debugger ^[1447]
4	Select the server process hosting the application	
5	<p>Click on the OK button</p> <p>A confirmation message displays in the Debug Output window, stating that the process has been attached</p>	

The breakpoints could show a question mark. In this case the Class might not have been loaded yet by the VM. If the question mark remains even after you are sure the Class containing the breakpoint has been loaded, then either:

- The binaries being executed by the server are not based on the source code
- The debugger cannot reconcile the breakpoint to a source file (check class paths), or
- The JVM has not loaded the Sparx Systems agent

Learn More:

- [JBoss Server](#)^[1427]
- [Apache Tomcat Server](#)^[1428]
- [Apache Tomcat Windows Service](#)^[1428]

Topics:

Topic	Detail	See also
Abstract	<p>Consider the JBoss example below; the source code for a simple servlet is located in the directory location:</p> <p><i>C:\Benchmark\Java\JBoss\Inventory</i></p> <p>The binaries executed by JBoss are located in the JAW.EAR file in this location:</p> <p><i>C:\JBoss\03b-dao\build\distribution</i></p> <p>The Enterprise Architect debugger has to be able to locate source files during debugging; to do this it also uses the CLASSPATH, searching in any listed path for a matching JAVA source file, so the CLASSPATH must include a path to the root of the package for Enterprise Architect to find the source during debugging</p> <p>The following is an excerpt from the command file that executes the JBoss server; the Class to be debugged is at:</p> <p><i>com/Inventory/dto/car DTO</i></p> <p>Therefore, the root of this path is included in the <i>JBoss</i> classpath</p>	
Example Code	<pre> RUN. BAT ----- set SOURCE=C: \ Benchmark\ Java\ JBoss\ Inventory set JAVAC_JAR=%JAVA_HOME%\lib\tools.jar if "%JBoss_CLASSPATH%" == "" (set JBoss_CLASSPATH=%SOURCE%; % JAVAC_JAR%; %RUNJAR%;) else (set JBoss_CLASSPATH=%SOURCE%; % JBoss_CLASSPATH%; %JAVAC_JAR%; %RUNJAR%;) set JAVA_OPTS=%JAVA_OPTS% -agentpath:"c: \program files\sparx systems\ssjavaprofiler75" </pre>	

Topics:

Topic	Detail	See also
Configuration	<p>This configuration is for the same application as outlined in the <i>JBOSS server</i> configuration topic</p> <p>The following two points are important:</p> <ul style="list-style-type: none"> The Java VM option: <ul style="list-style-type: none"> - agent path: c:\program files\sparx systems\ea\ssjavaprofiler75 The addition to the Class path property of the path to the source code: <ul style="list-style-type: none"> C:\JBOSS\03b-dao\common\src; <p><input checked="" type="checkbox"/> Use default</p> <p>Java Virtual Machine:</p> <p>C:\SunJava\jre 1.7.0\bin\client\jvm.dll</p> <p>Java Classpath:</p> <p>C:\Tomcat Local Source\src;C:\Java\jdk1.6.0_07\jre;C:\Java\jdk1.6.0_07</p> <p>Java Options:</p> <p>-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.util.logging.config.file=C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf\logging.properties -agentpath:C:\program files\sparx systems\ea\ssjavaprofiler75</p> <p>Tomcat Server Configuration Example</p>	JBOSS Server ^[1427]

Topics:

Topic	Detail	See also
Configuration	<p>For users running Apache Tomcat as a Windows™ service, it is important to configure the service to enable interaction with the Desktop; failure to do so causes debugging to fail within Enterprise Architect</p> <p>Log on as:</p> <p><input checked="" type="radio"/> Local System account</p> <p><input checked="" type="checkbox"/> Allow service to interact with desktop</p> <p>Select the Allow service to interact with desktop checkbox</p>	

11.2.6.2.4 .NET

This section describes how to configure Enterprise Architect for debugging .NET applications. It covers:

- [General Setup](#)^[1429]
- [Debugging Managed code in an Unmanaged Application](#)^[1429]
- [Debug COM Interop](#)^[1430]
- [Debug ASP.NET](#)^[1431]

11.2.6.2.4.1 General Setup for .NET

This is the general setup for debugging Microsoft .NET applications. You have two options when debugging:

- Debug an Application
- Attach to an application that is running

Option 1 - Debug an application

Field	Action	See also
Debugger	Select Microsoft .NET as the debugging platform	Debugging Unmanaged Code <small>[1429]</small> Debugging ASP.NET <small>[1431]</small>
Mode	Select the Run radio button	
Default Directory	This is set as the default directory for the process being debugged	
Application	Select and enter either the full or the relative path to the application executable	
Command Line Arguments	Parameters to pass to the application at startup	
Show Console	Create a console window for the debugger; not applicable to attaching to a process	
Symbol Search Path	Specify any additional paths to locate debug symbols for the debugger; separate the paths with a semi-colon	

Option 2 - Attach to an application that is running

Field	Action	See also
Debugger	Select Microsoft .NET as the debugging platform	Debugging Unmanaged Code <small>[1429]</small> Debugging ASP.NET <small>[1431]</small>
Mode	Select the Attach to Process radio button	

11.2.6.2.4.2 Debugging an Unmanaged Application

Topics:

Topic	Detail	See also
Usage	<p>If you are debugging managed code using an unmanaged application, the debugger might fail to detect the correct version of the Common Language Runtime (CLR) to load</p> <p>You should specify a config file if you don't already have one for the debug application specified in the Debug command of your script</p> <p>The config file should reside in the same directory as your application, and take the format:</p>	http://www.msdn2.microsoft.com/en-us/library/9w519wzk.aspx (Online Resource)

Topic	Detail	See also
	<p><code>name.exe.config</code> where <i>name</i> is the name of your application</p> <p>The version of the CLR you specify should match the version loaded by the managed code invoked by the debuggee</p> <p>The following is a sample config file:</p> <pre><configuration> <startup> <requiredRuntime version="version" /> </startup> </configuration></pre> <p><i>Version</i> is the version of the CLR targeted by your plugin or COM code</p>	

11.2.6.2.4.3 Debug COM Interop

Enterprise Architect enables you to debug .NET managed code executed using COM in either a Local or an In-Process server.

This feature is useful for debugging Plugins and ActiveX components.

How To:

Step	Action	See also
1	Create a package in Enterprise Architect and import the code to debug	Software Engineering 
2	Ensure the COM component is built with debug information	
3	Create a Script for the Package	
4	In the Debug tab, you can select to either attach to an unmanaged process or specify the path to an unmanaged application to call your managed code	
5	Add breakpoints in the source code to debug	

Topics:

Topic	Detail	See also
Attach to an Unmanaged Process	<p>If you are using:</p> <ul style="list-style-type: none"> An In-Process COM server, attach to the client process A Local COM Server, attach to the server process <p>Click on the Debug window Run button (or press (F6)) to display a list of processes from which you can choose</p>	

Notes:

- Detaching from a COM interop process you have been debugging terminates the process; this is a known issue for Microsoft .NET Framework, and information on it can be found on many of the MSDN .NET blogs

11.2.6.2.4.4 Debug ASP .NET

Topics:

Topic	Detail	See also
Usage	<p>Debugging for web services such as ASP requires that the Enterprise Architect debugger is able to attach to a running service</p> <p>Begin by ensuring that the directory containing the ASP .NET service project has been imported into Enterprise Architect and, if required, the web folder containing the client web pages</p> <p>If your web project directory resides under the website hosting directory, you can import from the root and include both ASP code and web pages at the same time</p> <p>It is necessary to launch the client first, as the ASP .NET service process might not already be running; load the client using your browser - this ensures that the web server is running</p> <p>In the debug setup you must select the Attach to process radio button, and then type the keyword Attach in the script; this keyword causes the debugger to prompt you for a process at runtime</p> <p>Click on the Debug window Run button to start the debugger; the Attach To Process dialog displays</p> <p>The name of the process varies across Microsoft operating systems, as explained in the <i>ASP .NET SDK</i>; for example, under Windows Vista the name of the IIS process is <i>w3wp.exe</i></p> <p>On Windows XP, the name of the process resembles <i>aspnet_wp.exe</i>, although the name could reflect the version of the .NET framework that it is supporting</p> <p>There can be multiple ASP.NET processes running under XP; you must ensure that you attach to the correct version, which would be the one hosting the .NET framework version that your application runs on; check the <i>web.config</i> file for your web service to verify the version of .NET framework it is tied to</p> <p>The Debug window Stop button should be enabled and any breakpoints should be red, indicating they have been bound</p>	<p>Start and Stop Debugger^[1447]</p> <p>Breakpoint and Marker Management^[1436]</p>

You can set breakpoints at any time in the web server code. You can also set breakpoints in the ASP web page(s) if you imported them.

Notes:

- Some breakpoints might not have bound successfully, but if none at all are bound (indicated by being dark red with question marks) something has gone out of sync; try rebuilding and re-importing source code

11.2.6.2.5 The PHP Debugger

The Enterprise Architect PHP Debugger enables you to debug PHP.exe scripts. This section discusses basic setup and the various debugging scenarios that are commonly encountered; the scenarios concern themselves with the mapping of file paths, which is critical to the success of a remote debugging session.

- Script Setup
- Local Windows Machine (Apache Server)
- Local Windows Machine (PHP.exe)
- Remote Linux Machine (Apache Server)
- Remote Linux Machine (PHP.exe)

Topics:

Topic	Detail	See also															
Script Setup	<p>An Analyzer Script is a basic requirement for debugging in Enterprise Architect; you create a script using the toolbar of the Execution Analyzer</p> <p>Select PHPXDebug as the debugging platform; when you select this platform the property page displays the following connection settings:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>host</td> <td>localhost</td> <td>The adaptor that Enterprise Architect listens on for from PHP</td> </tr> <tr> <td>port</td> <td>9000</td> <td>The port that Enterprise Architect listens on for the</td> </tr> <tr> <td>localpath</td> <td>%LOCAL%</td> <td>Specifies the local file path to be mapped to a remote file path This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the local file path</td> </tr> <tr> <td>remotepath</td> <td>%REMOTE%</td> <td>Specifies the remote file path that a local file path is mapped to This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the remote file path</td> </tr> </tbody> </table>	Field	Value	Meaning	host	localhost	The adaptor that Enterprise Architect listens on for from PHP	port	9000	The port that Enterprise Architect listens on for the	localpath	%LOCAL%	Specifies the local file path to be mapped to a remote file path This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the local file path	remotepath	%REMOTE%	Specifies the remote file path that a local file path is mapped to This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the remote file path	
Field	Value	Meaning															
host	localhost	The adaptor that Enterprise Architect listens on for from PHP															
port	9000	The port that Enterprise Architect listens on for the															
localpath	%LOCAL%	Specifies the local file path to be mapped to a remote file path This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the local file path															
remotepath	%REMOTE%	Specifies the remote file path that a local file path is mapped to This is a remote debugging setting; for local debugging the value is a placeholder; you should edit it to fit the remote file path															
Local Machine Apache Server	<p>In this situation, consider the following configuration:</p> <ul style="list-style-type: none"> • O/S: Windows7 • Network computer name: <i>MyPC</i> • Network share <i>MyShare</i> mapped to <i>c:\myshare</i> • Source files in Enterprise Architect have been imported from <i>c:\myshare\apache\myapp\scripts</i> • Apache document root is set to <i>//MyPC/MyShare/apache</i> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> • host: localhost • port: 9000 • localpath: <i>c:\myshare\apache\</i> • remotepath: <i>MyPC/MyShare/apache/</i> 																
Local Machine	<p>In this scenario an Analyzer Script for the connection parameters might be configured as follows, as file paths always map to same physical path:</p>																

Topic	Detail	See also
PHP.EXE	<ul style="list-style-type: none"> • host: localhost • port: 9000 • localpath: • remotepath: 	
Remote Linux Machine Apache Server	<p>In this situation consider the following configuration:</p> <ul style="list-style-type: none"> • Local Machine <ul style="list-style-type: none"> • O/S: Windows7 • Source files in Enterprise Architect have been imported from <i>c:\myshare\apache\myapp\scripts</i> • Remote Machine <ul style="list-style-type: none"> • O/S: Linux • Apache document root is set to <i>home/apache/htdocs</i> • Source files in Apache are located at <i>home/apache/htdocs/myapp/scripts</i> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> • host: localhost • port: 9000 • localpath: <i>c:\myshare\apache\</i> • remotepath: <i>home/apache/htdocs/</i> 	
Remote Linux Machine PHP.exe	<p>In this situation consider the following configuration:</p> <ul style="list-style-type: none"> • Local Machine <ul style="list-style-type: none"> • O/S: Windows7 • Source files in Enterprise Architect have been imported from <i>c:\myshare\apache\myapp\scripts</i> • Remote Machine <ul style="list-style-type: none"> • O/S: Linux • Source files in Apache located at <i>home/myapp/scripts</i> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> • host: localhost • port: 9000 • localpath: <i>c:\myshare\apache\</i> • remotepath: <i>home/</i> 	

Learn more:

- [PHP Debugger - System Requirements](#)^[1433]
- [PHP Debugger Checklist](#)^[1434]

11.2.6.2.5.1 PHP Debugger - System Requirements

This topic identifies the system requirements and operating systems for the Enterprise Architect PHP debugger.

System Requirements:

- Enterprise Architect version 9

- PHP version 5.3 or above
- PHP zend extension XDebug 2.1 or above
- For web servers such as Apache, a server version that supports the PHP version

Supported Operating Systems:

- Client (Enterprise Architect)
 - Microsoft Windows XP and above
 - Linux running Crossover Office
- Server (PHP)
 - Microsoft Windows XP and above
 - Linux

11.2.6.2.5.2 PHP Debugger Checklist

This topic provides a supplementary checklist and troubleshooting guide for debugging PHP scripts in Enterprise Architect.

Topics:

Topic	Detail		See also
System Requirements	<ul style="list-style-type: none"> • Apache HTTP Web Server version 2.2 • PHP version 5.3 or above • XDebug version 2.1.1 		http://httpd.apache.org/ http://windows.php.net/ http://www.xdebug.org/download.php
Checklist	Enterprise Architect	<ul style="list-style-type: none"> • The model has an Analyzer Script configured to use the PHP-XDebug platform • PHP source code has been imported into the model (for recording and testpoints) • When the PHP-XDebug platform is selected from the Analyzer Script dialog, default runtime settings are listed in the Connection field <ul style="list-style-type: none"> • local path: %LOCAL% • remotepath: %REMOTE% <p>Either define local paths for these default variables or edit the script to provide actual paths</p> <p>For example: local source, remote source</p> <p><i>localpath:c:\code samples\vea\php\sample</i> <i>remotepath:web server/sample</i></p> <p>where:</p> <ul style="list-style-type: none"> • <i>web server</i> is a network or local share • <i>sample</i> is a folder below share 	
	PHP	<p>In order to debug PHP scripts in Enterprise Architect, it is a requirement that the PHP is configured properly to load the XDebug extension</p> <p>Settings similar to those below should be used:</p> <ul style="list-style-type: none"> • [xdebug] • xdebug.extended_info=1 	

Topic	Detail	See also
	<ul style="list-style-type: none"> xdebug.idekey=ea xdebug.remote_enable=1 xdebug.remote_handler=dbgp xdebug.remote_autostart=1 xdebug.remote_host=X.X.X.X xdebug.remote_port=9000 xdebug.show_local_vars=1 <p>The IP address X.X.X.X refers to and should match the host specified in the model Analyzer Script</p> <p>The IP address is the address XDebug connects with and the same address the Enterprise Architect PHP agent listens on</p>	
	<p>Apache</p> <p>For debugging using Apache, the following lines should be present in the Apache configuration file, <i>httpd.conf</i></p> <pre>LoadModule php5_module "php_home/ php5apache2_2.dll" AddHandler application/x-httpd-php . php PHPLibraryDir "php_home"</pre> <p>where <code>php_home</code> is the PHP installation path (the path where <code>php.ini</code> and <code>apache.dll</code> exist)</p>	
Troubleshooting	<p>To prevent both PHP and Apache timeouts during a debugging session, the settings below might require modification</p> <p>These settings were used while developing the PHP Debugging agent in Enterprise Architect</p>	
	<p>PHP</p> <p>File: php.ini</p> <pre>; EA prevent PHP timeouts when debugging PHP extensions max_execution_time = 0 ; EA prevent web server timeouts when debugging PHP extensions max_input_time = -1 ; EA log errors display_errors = On ; EA display startup errors display_startup_errors = On</pre>	
	<p>Apache</p> <p>File: httpd.conf</p> <pre>; EA prevent timeouts while debugging php extensions Timeout 60000</pre>	

11.2.6.3 Tracepoint Output

The Tracepoints page of the Analyzer Script enables you to direct where the output from any Trace statements goes during a debug session.

Field	Detail	See also
Output	<p>You can select from two options:</p> <ul style="list-style-type: none"> Screen (Default) - The output is directed to the Debugger 	

Field	Detail	See also
	Window <ul style="list-style-type: none"> • File - The output is directed to file 	
Folder	Enter the folder to use for Trace statement log files	
Filename	Enter the name to use for the Trace statement log files	
Overwrite	If selected, the specified file is overwritten each time a debug session is started	
Auto Number	If selected, the Trace log file is composed of the filename you specify and a number Each time you start a Debug session, the number is incremented	
Prefix trace output with function	If selected, any trace statements executed during the Debug session run are prefixed with the current function call	

11.2.6.4 Breakpoint and Marker Management

Breakpoints work in Enterprise Architect much like in any other debugger. Adding a breakpoint notifies the debugger to trap code execution at the point you have specified. When a breakpoint is encountered by a thread of the application being debugged, the source code is displayed in an editor window, and the line of code where the breakpoint occurred is highlighted.

Selecting a different package in the project affects which breakpoints are displayed. An Enterprise Architect model maintains breakpoints for every package having a *Build Script - Debug* command.

Access: [Analyzer | Breakpoints & Markers > Breakpoints & Markers](#)

The Breakpoint and Markers Toolbar:

Button	Description
	Enables all breakpoints
	Deletes all breakpoints
	Disables all breakpoints
	Limits any recording to specified relative stack depth
	Current breakpoint set. Default set is normally used for debugging and is personal. Other sets are shared between users of Model.
	Menu allowing management of breakpoint sets.
	Changes how breakpoints are grouped. You can select no grouping, group by file or group by class.

Breakpoint States:

DEBUGGER STATE			
	Running	Not running	See also
	Bound	Enabled	
	Disabled	Disabled	
	Not bound - this usually means that the DLL is not yet loaded or was not built with debug information	N/a	
	Failed - this usually means a break could not be set at this time	N/a	Failure to Bind Breakpoint ^[1439]

11.2.6.4.1 Breakpoint Properties

Breakpoints in Enterprise Architect support a number of additional properties that change when a breakpoint is hit, and determine what happens when it is hit.

Access: [Breakpoint Context Menu | Properties](#)

Use to:

- Change breakpoint actions.
- Move the breakpoint to another line.
- Add constraints to restrict when a breakpoint is hit.
- Add information to be logged when the breakpoint is hit.

Reference:

Field	Usage	See also
Action	Change the behavior when the breakpoint is hit	Marker Types ^[1642]
Line	Change the line that this breakpoint is on	
Constraints	Associate a condition with the breakpoint The program is allowed to run if the condition is not met when the breakpoint is encountered	
Log	Log information to the Debug Output window when the breakpoint is hit	

Specifying constraints:

You can compose constraints using any variables that are in scope at the time the breakpoint occurs. For example:

```
m_strName="John"
```

```
m_strName=""
```

```
m_nCount > 0
```

You can also join constraints using logical operands, for example:

```
m_strName="John" AND m_nCount > 0
```

You can access elements of an array or pointer, for example:

```
m_pNames[10]="John"
```

You can also nest constraints using parentheses to impose precedence, for example:

```
( x>0 AND y>0) OR (x < 0 AND y < 0)
```

11.2.6.4.2 Setting Code Breakpoints

Normal Breakpoints are typically set on a line of source code. When the Debugger hits the indicated line during normal execution, the Debugger halts execution and displays the local variables, call stack, threads and other run-time information.

How To:

To set a breakpoint on a line of code, follow the steps below:

Step	Action	See also
1	Open the source code to debug in the integrated source code editor;	
2	<p>Find the appropriate code line and click in the left margin column. A solid red circle in the margin indicates that a breakpoint has been set at that position.</p> <p>If the code is currently halted at a breakpoint, that point is indicated by a blue arrow next to the marker.</p>	

11.2.6.4.3 Setting Data Breakpoints

Data breakpoints enable the debugger to halt execution when the value of a pre-determined memory variable is changed. This can be useful when trying to track down the point at which a variable is modified during program execution, especially if it is not clear how program execution is affecting a particular object state.

Topics:

Topic	Detail	See also
Setting Data Breakpoints	<p>Data breakpoints can currently only be set by right-clicking on the variable in the Locals window and selecting the Break When Variable is Modified context menu option</p> <p>This means that in order to establish a data breakpoint you must first set a normal breakpoint at a point in the code that presents the required scope of local variables to choose from</p> <p>Once the program has halted execution, locate the variable of interest and select the Break When Variable is Modified context menu option</p>	<p>Viewing the value of local variables ^[1442]</p> <p>Breakpoint and Marker Management ^[1438]</p>

11.2.6.4.4 Failure to Bind BreakpointTopics:

Topic	Detail	See also
Breakpoint Failures	<p>A breakpoint failure occurs if there is a problem in binding the breakpoint</p> <p>A warning message displays in the Details column of the Breakpoints & Markers window, identifying the type of problem; for example:</p> <ul style="list-style-type: none"> • The source file for the breakpoint does not match the source file used to build the application image • The time date stamp on the file is greater than that of the image <p>A warning message is also output to the Debug Output window</p>	

11.2.6.4.5 Delete and Disable BreakpointsTopics:

Topic	Detail	See also
Delete, Disable and Enable Breakpoints	<p>To delete a specific breakpoint, either:</p> <ul style="list-style-type: none"> • If the breakpoint is enabled, click on the red breakpoint circle in the left margin of the Source Code Editor • Right-click on the breakpoint marker in the editor and select the appropriate context menu option, or • Select the breakpoint in the Breakpoints & Markers tab and press (Delete) <p>Whether you are viewing the Breakpoints folder or the Breakpoints & Markers window, you can right-click on an existing breakpoint and select a context menu option either to delete it or to convert it to a start recording marker or end recording marker</p>	<p>Marker Types ^[1642]</p>

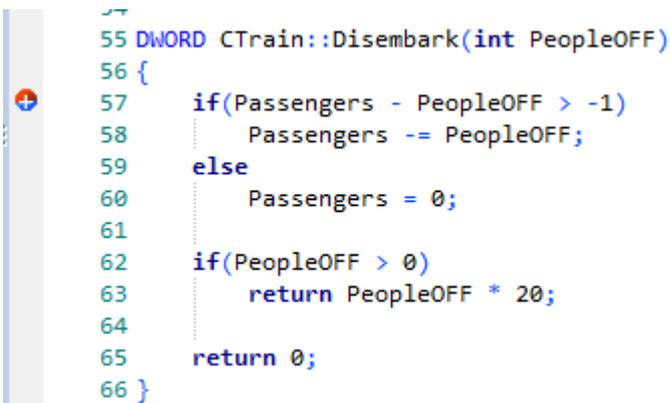
Topic	Detail	See also
	<p>You can also delete all breakpoints by clicking on the Delete all breakpoints button on the Breakpoints & Markers window toolbar</p> <p>()</p> <p>To disable a breakpoint, deselect its checkbox on the Breakpoints & Markers window or, to disable all breakpoints, click on the Disable all breakpoints button in the toolbar (); the breakpoint is then shown as an empty grey circle</p> <p>Select the checkbox or use the Enable all breakpoints button to enable it again ()</p>	

11.2.6.4.6 Trace Statements

Tracepoints are set in the code editor. Like breakpoints, they are placed on a line of code. When that line of code executes, the Execution Analyzer evaluates and prints the statement either to the Debugger output window or to file, if this has been configured in the Analyzer Script.

How to:

To set Tracepoints, follow the steps below:

Step	Action	See also
1	Open the source code to debug in the integrated source code editor	
2	Find the appropriate code line and right click in the left margin to bring up the Analyzer context menu Select the Add Tracepoint Marker option	
3	 <pre> 55 DWORD CTrain::Disembark(int PeopleOFF) 56 { 57 if(Passengers - PeopleOFF > -1) 58 Passengers -= PeopleOFF; 59 else 60 Passengers = 0; 61 62 if(PeopleOFF > 0) 63 return PeopleOFF * 20; 64 65 return 0; 66 } </pre>	
4	Right-click on the Tracepoint and choose the Properties context menu option	
5	In the Log field type the statement	

Specifying Tracepoints:

Using the example in the image above, we could output the number of people getting off a train by using the following statement:

```
There were @Passengers before @PeopleOFF got off the train
```

To print a variable, precede the variable name with the @ token. If a variable represents a string you should precede the variable with a \$ token.

11.2.6.4.7 How Markers are Stored

Topics:

Topic	Detail	See also
Storage	The Default always exists for a Model. It is a personal workspace and is not shared. It is stored externally. Marker Sets are stored in the model and are available to all users of the Model. They are shared,	

11.2.6.5 Debugging Actions

This section describes the actions you perform in running a debug session. It covers:

- [Starting and Stopping the Debugger](#) ^[1447]
- [Debugging a Subsequent Process](#) ^[1448]
- [Stepping Over Lines of Code](#) ^[1448]
- [Stepping Into Function Calls](#) ^[1449]
- [Stepping Out of Functions](#) ^[1449]
- [View the Call Stack](#) ^[1441]
- [Viewing the Local Variables](#) ^[1442]
- [Viewing the Content of Long Strings](#) ^[1443]
- [Viewing Variables in Other Scopes](#) ^[1444]
- [Inspecting Process Memory](#) ^[1445]
- [Setting Breaks for When a Variable Changes Value](#) ^[1445]
- [Showing Loaded Modules](#) ^[1446]
- [Showing Output from the Debugger](#) ^[1446]
- [Debugging Tooltips in Code Editors](#) ^[1447]

11.2.6.5.1 View the Call Stack

The Call Stack window is used to display all currently running threads in a process. It can be used to identify which thread is operational immediately before program failure occurs. A Stack trace is displayed whenever a thread is suspended, through one of the step actions or through encountering a **breakpoint**. The Call Stack can record a history of Stack changes along with the ability to generate Sequences Diagrams from this history.

Access: Analyzer | Call Stack

Use To:

- View Stack history to understand the execution of a process
- View Threads
- Save a Call Stack for later use
- Record Call Stack changes for Sequence Diagram generation

- Generate a Sequence Diagram from the Call Stack
- View the related code line in Source Code Editor

Topics:

Topic	Icon	Detail	See also
Indicators		<ul style="list-style-type: none"> • A green or yellow arrow highlights the current stack frame • A blue arrow indicates a thread that is running • A red arrow indicates a thread for which a stack trace history is being recorded 	
Actions		<p>Toolbar Actions:</p> <ul style="list-style-type: none"> • Save a Call Stack to file • Generate a Sequence diagram from the current Stack • Copy Stack to Recording history - adds the stack details to the Record History for Sequence Diagram generation • Toggle Stack Depth - show either the full Stack View or only frames with source • Double-clicking a thread/frame takes you to the related line of code in the Source Code Editor; local variables are also refreshed for the selected frame <p>Right-click on an items in the Stack list and select context menu options to:</p> <ul style="list-style-type: none"> • Record Stack changes • Auto-Record • Stop the Recording • Generate a Sequence Diagram of the Stack • Copy Stack to Record History 	
Toolbar		Save Stack to file	
		Generate Sequence diagram from Stack	
		Copy Stack to recording history	
		Toggle Stack View	
		Stop recording	

Learn More:

- [Breakpoint and Marker Management](#)^[1436]

11.2.6.5.2 View the Local Variables

Whenever a thread encounters a breakpoint, this window displays all the local variables for the thread at its current stack frame.

The value and the type of any in-scope variables are displayed in a tree. Local variables are displayed with colored box icons with the following meanings:

Color	Meaning
Blue	Object with members

Color	Meaning
Green	Arrays
Pink	Elemental types
Yellow	Parameters
Red	Workbench Instance

Learn More:

- [Breakpoint and Marker Management](#)^[1436]
- [View the Call Stack](#)^[1441]

11.2.6.5.3 View Content Of Long Strings**Topics:**

Topic	Detail	See also
Debugging Actions - View Entire Content Of Long Strings	For efficiency, the Locals window only shows partial strings; the size of any variable value displayed in the window can be up to 256 characters To view the entire value of a variable, right-click on it and select the View in Editor context menu option; the String Viewer dialog displays	

11.2.6.5.4 View Elements of Array**Access: Analyzer | Watches**

It is not always possible to evaluate the content of large data structures or arrays using the local variables display. To inspect one or more elements of an array you can use the Watches window. In the toolbar edit box you type the name of the variable followed by the start element and the number of elements to display. The start element is enclosed in square brackets and the count of elements is enclosed in parentheses, as follows:

```
variable[ start_element ] ( count_of_elements )
```

m_pStrings[0](10)	Displays ten elements of string array starting at element zero.	char* m_pStrings;
m_Numbers[100]	Displays the 100th element of integer array m_Numbers	int m_Numbers[200]

11.2.6.5.5 View Variables in Other Scopes

Topics:

Topic	Detail	See also
Watches	<p>The Watches window is most useful for native code (C, C++, VB) where it can be used to evaluate data items that are not available as Local Variables - data items with module or file scope and static Class member items</p> <p>You can also use the window to evaluate static Class member items in Java and .NET</p> <p>To use the Watches window, type the name of the variable to examine in the field in the window toolbar, and either press (Enter) or click on the Add new watched item icon</p> <p>To examine a static Class member variable in C++, Java or Microsoft .NET, enter its fully qualified name; for example:</p> <pre>CMyCl ass : : MySt at i cVar</pre> <p>To examine a C++ data symbol with module or file scope, just enter its name</p> <p>Variables are evaluated by looking at the current scope; that is, the module of the current stack frame (you can change the scope at a breakpoint by double-clicking the frame in the call stack)</p> <p>If the global variable exists in a different module, you can examine the variable by prefixing the module name to the variable; for example:</p> <pre>modul ename! var i abl e_ name</pre> <p>The names of the items to evaluate persist for the package and user ID, so the next time <i>you</i> debug the same project, the items evaluate automatically whenever a breakpoint occurs; they do not appear if another user debugs the same code</p> <p>If necessary, you can delete items using the Delete all watched items icon in the toolbar, or the right-click context menu option inside the window</p>	<p>View the Local Variables ^[1444]</p>

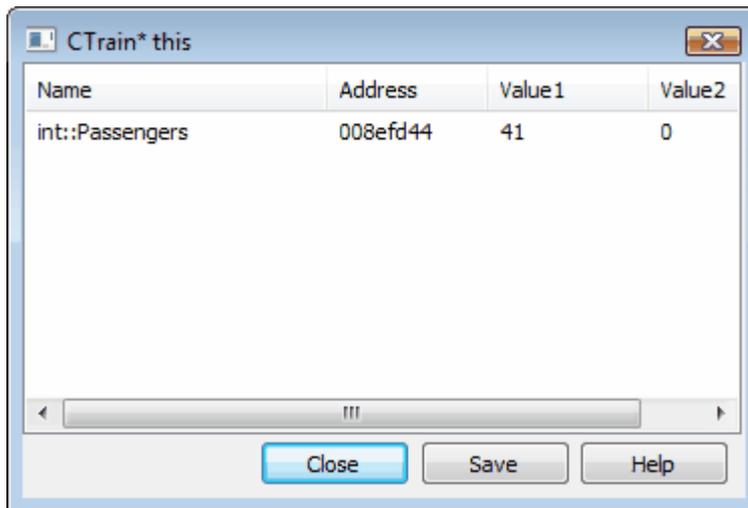
11.2.6.5.6 Variable Snapshots

Snapshots allow you to compare the value of an object at one point in time with its value at a later point in time. You can take a snapshot of a variable when the program is at a breakpoint. The debugger copies the values of not only the selected variable but any of its members and and so on until it can find no debug information for a member or no more members can be found.

The procedure for using variable snapshots is typically as follows:

1. Set two breakpoints (called **start** and **end**) The start is usually placed at the start of a function and the end breakpoint at the end of the function.
2. At the **start** breakpoint select a variable in the Locals window and select the **Save Variable Snapshot** context menu option.
3. Run the program.
4. When the **end** breakpoint is reached, select the **Compare Variable Snapshots** context menu option; a dialog displays that details the original value from the first snapshot and the current value from the second snapshot (see the example below).

The following image shows the comparison between values of a variable, captured using this feature. (This is taken from the *EA.Example* model.)



11.2.6.5.7 Inspect Process Memory

Access: [Analyzer](#) | [More Analyzer Windows](#) | [Memory Viewer](#)

Topics:

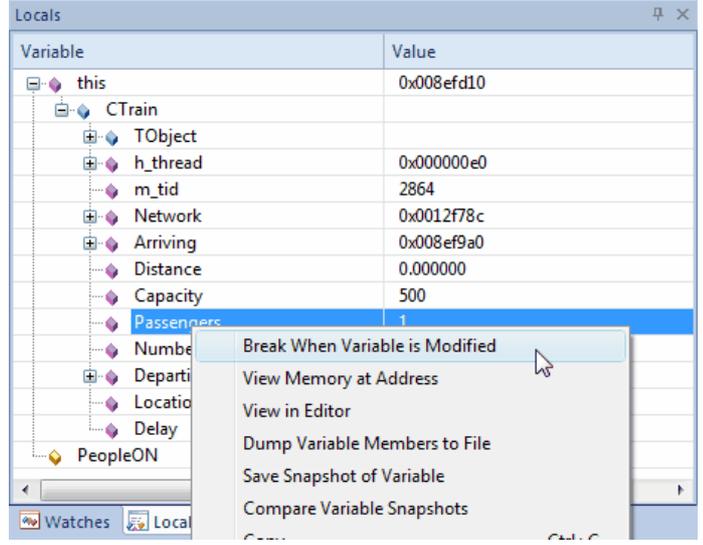
Topic	Detail	See also
Memory Viewer	<p>You can display the raw values at a memory address or for a variable in a window using the Memory Viewer - select the View Memory at Address context menu option</p> <p>The Memory Viewer displays the raw values at a memory address</p> <p>The Memory Viewer is available for debugging Microsoft Native Code Applications (C, C++, VB) running on Windows or within WINE on Linux</p>	

11.2.6.5.8 Break When a Variable Changes Value

You can set a breakpoint on a variable such that when it is modified the debugger halts at the line of code that has just caused the memory to be changed. When such a breakpoint occurs the code editor is displayed and the line highlighted. The log in the Debugger window also prints a statement to the effect.

Topics:

Topic	Detail	See also
Usage	<p>An invalid or uninitialised variable can cause the program behaviour to differ from expected. This tool enables you to halt execution whenever a certain variable has its value changed.</p> <p>The example below creates a notification on a variable from the Locals window. The item being watched is the Passengers integer member of the CTrain class.</p>	

Topic	Detail	See also
		

Notes:

- This feature is not presently supported by the Microsoft .NET platform.

11.2.6.5.9 Show Loaded Modules

The debugger **Modules** window lists the modules loaded by the process being debugged.

The columns on this window are described below:

Column	Action
Path	Determine the file path of the loaded module.
Modified Date	Determine the local file date and time the module was modified.
Debug Symbols	Establish the debug symbols type, whether debug information is present in the module and whether line information is present for the module (required for debugging).
Symbol File Match	Check the validity of the symbol file; if the value is false, the symbol file is out of date.
Symbol Path	Determine the file path of the symbol file, which must be present for debugging to work.
Modified Date	Determine the local file date and time the symbol file was created.

11.2.6.5.10 Show Output from Debugger**Topics:**

Topic	Detail	See also
Usage	During a debug session the Debugger emits messages	

Topic	Detail	See also
	<p>detailing both startup and termination of session, to its Output tab. Details of exceptions and any errors are also output to this tab. Any trace messages such as those output using <i>Java System.out</i> or <i>.NET System.Diagnostics.Debug</i> are also captured and displayed here.</p> <p>If you double-click on a debug message, either:</p> <ul style="list-style-type: none"> • A pop-up displays with more complete message text, or • If there has been a memory leak, the file is displayed at the point at which the error occurred 	

11.2.6.5.11 Debug Tooltips in Code Editors

Topics:

Topic	Detail	See also
Inspecting Member Variables	<p>During debugging, whenever a thread is suspended at a line of execution, you can inspect member variables in the Editor window.</p> <p>To evaluate a member variable, use the mouse to move the cursor over the variable in the Editor window, as shown in the following examples.</p> <pre> public void Print() { int n = 0; while (names[n].Length > 0) { names = { [4] names[0]=book, names[0]=book, names[1]=novel, names[2]=film, ... } Document d = new Document (names[n++]); d.Print(); } } public void Print() { int n = 0; while (32-bit signed integer n=0 0) { Document d = new Document (names[n++]); d.Print(); } } </pre>	

11.2.6.5.12 Start & Stop Debugger

Topics:

Topic	Detail	See also
Usage	<p>If Basic Setup has been completed, pressing (F6) starts the application using the configured Debugger.</p> <p>If not, debugging is still possible by using the Attach button on either one of the Debugger toolbars.</p> <p>To stop debugging, click on the Stop button  in the Debug</p>	Setup ^[1392]

Topic	Detail	See also
	window toolbar, or press (Ctrl+Alt+F6) .	

Notes:

- In most situations, the debugger ends:
 - when it encounters breakpoints (which should be set beforehand)
 - when the debug process terminates
 - when the Java Class thread exits
- However, due to the nature of the Java Virtual Machine, it is necessary at times for Java developers to stop the debugger manually with the **Stop** button

11.2.6.5.13 Debug Another Process**Topics:**

Topic	Detail	See also
Usage	<p>When debugging another process, the Attach To Process dialog is displayed.</p> <p>You can limit the processes displayed using the radio buttons at the top of the dialog. To find a service such as Apache Tomcat or ASP.NET, select the System radio button.</p> <p>You must choose the debugger when you select a process. However, if the selected Package has already been configured for debugging then the Debugger listed is the one specified in the Script.</p> <p>Once Enterprise Architect is attached to the process, any breakpoints encountered are detected by the debugger and the information is available in the Debugger windows.</p> <p>To detach from a process, click on the Debug Stop button.</p>	

11.2.6.5.14 Step Over Lines of Code**Topics:**

Topic	Detail	See also
Stepping Over	<p>You can step over the lines of a function using the Debug toolbar buttons.</p> <p>When you step to the end of the function, you step back to the caller.</p>  <p>Alternatively, press (Alt+F6) or select the Analyzer Debug Step Over context menu option.</p>	

11.2.6.5.15 Step Into Function Calls

Topics:

Topic	Detail	See also
Stepping In	<p>The <i>Step In</i> function is executed by clicking on the Step In button.</p>  <p>Alternatively, press (Shift+F6) or select the Analyzer Debug Step In context menu option.</p> <p>If no source is available for the function then the debugger continues stepping until it either enters a new function or reaches the next line of the current one.</p>	

11.2.6.5.16 Step Out of Functions

Topics:

Topic	Detail	See also
Stepping In	<p>The <i>Step Out</i> function is executed by clicking on the Step Out button.</p>  <p>Alternatively, press (Ctrl+F6) or select the Analyzer Debug Step Out menu option.</p> <p>If no source is available for the function then the debugger continues stepping until it either enters a new function or reaches the next line of the current one.</p>	

11.2.6.6 Recording Actions

This section describes how to perform the following debug recording actions:

- [Step through function calls](#) ^[1449]
- [Create a Sequence diagram of the Call Stack](#) ^[1450]
- [Save the Call Stack](#) ^[1450]

11.2.6.6.1 Step Through Function Calls

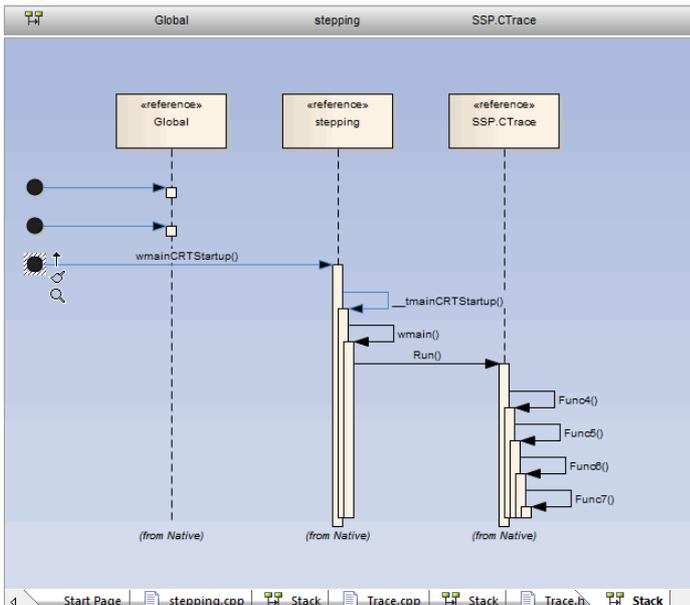
Topics:

Topic	Detail	See also
Stepping Through	<p>The Step Through function can be executed by clicking on the Step Through button on the Record & Analyze window toolbar</p>  <p>Alternatively, press (Shift+F6) or select the Analyzer Debug Step Into context menu option</p> <p>The <i>Step Through</i> command causes a <i>Step Into</i> command to</p>	

Topic	Detail	See also
	<p>be executed. If any function is detected, then that function call is recorded in the History window. The debugger then steps out, and the process can be repeated.</p> <p>This button enables you to record a call without having to actually step into a function. The button is only enabled when at a breakpoint and in manual recording mode.</p>	

11.2.6.6.2 Create Sequence Diagram of Call Stack

Topics:

Topic	Detail	See also
Usage	<p>To generate a Sequence diagram from the current Stack, click on the Generate Sequence Diagram of Stack button on the Call Stack window toolbar.</p> <p>This immediately generates a Sequence diagram in the Diagram View.</p> 	

11.2.6.6.3 Saving the Call Stack

On the **Call Stack** window, you can save the current Stack to file or copy the Stack to the recording history.

Toolbar:

Icon	Detail	See also
	Save Stack to file	
	Copy Stack to recording history	

11.2.7 Run

This section describes how to create a **command for running** your executable code.

Learn More:

- [Add run command](#)^[1451]

11.2.7.1 Add Run Command

This topic explains how you enter a command for running your executable.

Topics:

Topic	Detail	See also
Usage	This is the command that is executed when you select the Analyzer Run menu option. At its simplest, the script would contain the location and name of the file to be run.	
Example	<p>The following two examples show scripts configured to run a .Net and a Java application in Enterprise Architect.</p> <p>.Net:</p> <pre>C: \ benchmark \ cpp \ example_net_1 \ release \ example.exe</pre> <p>Java:</p> <pre>customer</pre> <p>The command listed in this field is executed as if from the command prompt. As a result, if the executable path or any arguments contain spaces, they must be surrounded in quotes.</p>	

Notes:

- Enterprise Architect provides the ability to start your application normally OR with debugging from the same script. The **Execution Analyzer** menu has separate options for starting a normal run and a debug run

11.2.8 Testing

This section describes how to create a **command for performing unit testing** on your code.

Learn More:

- [Add Testing Command](#)^[1451]

11.2.8.1 Add Testing Command

This topic explains how to configure the Test Page of an Analyzer Script to performing unit testing on your code.

Topics:

Topic	Detail	See also
Usage	<p>A sample script would contain a line to execute the testing tool of your choice, with the filename of the executable produced by the Build command as the option. To execute this test select the Analyzer Test Run Test Script menu option.</p> <p>Testing could be integrated with any test tool using the command line provided, but in these examples you can see how to integrate <i>NUnit</i> and <i>JUnit</i> testing with your source code. Enterprise Architect provides an inbuilt MDA Transform from source to Test Case, plus the ability to capture <i>xUnit</i> output and use it to go directly to a test failure. <i>xUnit</i> integration with your model is now a powerful means of delivering solid and well-tested code as part of the complete model-build-test-execute-deploy life-cycle.</p> <p>NUnit and JUnit must be downloaded and installed prior to their use. Enterprise Architect does not include these products in the base installer.</p>	
Execute Command As:	<p>Process</p> <p>Enter path to a program or batch file to run followed by any parameters.</p> <p>Batch File</p> <p>When using this option, you can enter multiple commands. These commands are then executed as a single script in a command console. You have access to any environment variables available in a standard command console.</p>	
Parse Output	<p>Selecting an Output Parser causes output from the command(s) to be captured to the Output window. The Output Parser field specifies the syntax to be expected. Double-clicking on a result in the Output window opens the corresponding code segment in Enterprise Architect's code window.</p> <p>Selecting the Build First checkbox ensures that the package is recompiled each time you run the test.</p> <p>If you include the string %N in your test script it is replaced by the fully namespace-qualified name of the currently selected Class when the script is executed.</p>	
Example	<p>NUnit</p> <pre>" C: \ Pr o g r a m F i l e s \ N U n i t \ b i n \ n u n i t - c o n s o l e . e x e " p i n \ d e b u g \ C a l c u l a t o r . e x e "</pre> <p>JUnit</p> <pre>j a v a j u n i t . t e x t u i . T e s t r u n n e r %N</pre> <p>The command listed in this field is executed as if from the command prompt. As a result, if the executable path or any arguments contain spaces, they must be surrounded in quotes.</p>	

11.2.8.2 Testpoints Output

The Testpoints page of the Analyzer Script allows you to configure the output of a Testpoint run.

Topic	Detail	See also
Output	You can select from two options: <ul style="list-style-type: none"> • Screen (Default) - The output is directed to the Testpoints tab of the System Output window • File - The output is directed to file 	Testpoint Management ^[1672]
Folder	Enter the folder to use for Testpoint log files.	
Filename	Enter the name to use for the Testpoints log files	
Overwrite	When Overwrite is selected, the file specified is overwritten each time a Testpoint run is performed.	
Auto Number	When this option is selected, the Testpoint output is composed of the filename you specify and the number of the Test run. Each time you perform a Test run the number is incremented.	
Prefix trace output with function	When this option is selected, any trace statements executed during the Testpoint run are prefixed with the current function call.	

11.2.9 Deploying

This section describes how to create a **command for deploying** the current package.

Learn More:

- [Add Deploy Command](#)^[1453]

11.2.9.1 Add Deploy Command

This topic explains how to configure the Deploy Page of an Analyzer Script to deploy your project.

Topics:

Topic	Detail	See also
Usage	This topic enables you to create a command for deploying a package. This script is executed when you select the Analyzer Deploy or by pressing Ctrl + Shift + Alt + F12	
Execute Command as:	Process If the deployment is handled externally, enter the path to the program or batch file to run followed by any parameters. This program is launched in a separate process. Example: C:\apache-ant-1.7.1\bin\ant.cmd myproject deploy	

Topic	Detail	See also
	<p>Batch File</p> <p>When using this option, you can enter multiple commands. These commands are then executed as a single script in a command console. You have access to any environment variables available in a standard command console.</p> <p>Example:</p> <pre>@echo on IF NOT EXIST "%1%" GOTO DEPLOY_NOWAR IF "%APACHE_HOME%" == "" GOTO DEPLOY_NOAPACHE xcopy /L "%1%" "%APACHE_HOME%\webapps" GOTO DEPLOY_END rem rem NO WAR FILE rem :DEPLOY_NOWAR echo "%1% WAR file not found" GOTO DEPLOY_END rem rem NO APACHE ENVIRONMENT VARIABLE rem :DEPLOY_NOAPACHE echo "APACHE_HOME environment variable not found" :DEPLOY_END pause</pre>	
Parse Output	<p>Selecting a Parser from the list causes output of the deploy script to be captured. The output is parsed according to the syntax selected from the list.</p> <p>The output window can be displayed by selecting View System Output</p>	

11.2.10 Searching Files

This topic describes how to use the File Search facility to search files for data names and structures.

Learn More:

- [Find in Files](#) ^[1454]

11.2.10.1 Find in Files

This topic describes the **File Search** control.

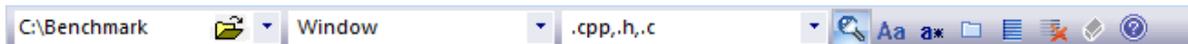
Topics:

Topic	Detail	See also
Usage	<p>File Text Searches are provided by the Search Window and from within the Code Editors.</p> <p>The Search window enables you to search for text in code files and scripts. You can select to display the results of the search in one of two formats:</p> <ul style="list-style-type: none"> • List View - each result line consists of the file path and line number, followed by the line text; multiple lines from one file are listed as separate entries 	

Topic	Detail	See also
	<ul style="list-style-type: none"> • Tree View - each result line consists of the file path that matches the search criteria, and the number of lines matching the search text within that file; you can expand the entry to show the line number and text of each line 	

Search Toolbar:

You can use the toolbar options in the **Search** window to control the search operation. The state of all buttons persists over time to always reflect your previous search criteria.



The options, from left to right, are as follows:

Reference:

Field	Usage	See also
Search Path list box	<p>Specify the folder to search.</p> <p>You can type the path to search directly into the text box, or click on the folder icon to browse for the path. Any paths you enter are automatically saved in the drop-down list, up to a maximum of ten; paths added after that overwrite the oldest path in the list.</p> <p>A fixed option in the drop-down list is Search in Scripts, which sets the search to operate on all local and user-defined scripts in the Scripts tab of the Scripting window. This option disables the Search File Types list box.</p>	
Search Text list box	<p>Specify the text to look for.</p> <p>You can type the text directly into the text box or click on the drop-down arrow to select from a previous entry in the list. The search text you enter is automatically saved in the list when you click on the Search button.</p> <p>The list box saves up to ten search queries. Search queries added after that overwrite the oldest query in the list.</p>	
Search File Types list box	<p>Limit the search to specific types of files. You can select multiple file types in a string, separated by either a comma or a semi-colon as shown in the image above.</p>	
Search button	<p>Begin the search.</p> <p>During the course of the search all other buttons in the toolbar are disabled. You can cancel the search at any time by clicking on the Search button again.</p> <p>If you switch any of the toggle buttons below, you must run the search again to change the output.</p>	
Case Sensitivity button	<p>Toggle the case sensitivity of the search. The tooltip message identifies the current status of the button.</p>	
Word Match button	<p>Toggle between searching for any match and searching for only</p>	

Field	Usage	See also
	those matches that form an entire word. The tooltip message identifies the current status of the button.	
SubFolders button	Toggle between limiting the search to a single path and including all subfolders under that path. The tooltip message identifies the current status of the button.	
Result View button	Select the presentation format of the search results - List View or Tree View format.	
Clear Results button	Clear the results.	
Clear Search Criteria button	Remove all the entries in the Search Path , Search Text and Search File Types list boxes, if required.	
Help button	Display this Help topic.	

11.3 Modeling Conventions



In order to get the most out of the round trip engineering in Enterprise Architect, you must be familiar with the modeling conventions used when generating and reverse engineering the languages you use. This topic describes the stereotypes, Tagged Values and other conventions used in code engineering in Enterprise Architect for the following supported languages:

Topic	Link
Action Script	ActionScript ^[1458]
Ada 2005 (Systems Engineering and Ultimate editions)	Ada 2005 ^[1459]
C	C ^[1461]
C#	C# ^[1463]
C++	C++ ^[1465]
Delphi	Delphi ^[1469]
Java	Java ^[1470]
PHP	PHP ^[1472]
Python	Python ^[1473]
SystemC(Systems Engineering and Ultimate editions)	SystemC ^[1474]
Verilog (Systems Engineering and Ultimate editions)	Verilog ^[1477]
VHDL(Systems Engineering and Ultimate editions)	VHDL ^[1478]
Visual Basic	Visual Basic ^[1480]
Visual Basic .NET	Visual Basic .NET ^[1475]

Notes:

- Enterprise Architect incorporates a number of visibility indicators or scope values for its supported languages. These include, for:
 - All languages - Public (+), Protected (#) and Private (-)
 - Java - Package (~)
 - Delphi - Published (^)
 - C# - Internal (~), Protected Internal (^)
 - ActionScript - Internal (~)
 - VB.NET - Friend (~), Protected Friend (^)
 - PHP - Package (~)
 - Python - Package (~)
 - C - Package (~)
 - C++ - Package (~)

11.3.1 ActionScript Conventions

Enterprise Architect supports round trip engineering of ActionScript 2 and 3, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
literal	Operation	A literal method referred to by a variable.
property get	Operation	A read property.
property set	Operation	A write property.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
dynamic	Class or Interface	The <i>dynamic</i> keyword.
final	ActionScript 3: Operation	The <i>final</i> keyword.
intrinsic	ActionScript 2: Class	The <i>intrinsic</i> keyword
namespace	ActionScript 3: Class, Interface, Attribute, Operation	The namespace of the current element.
override	ActionScript 3: Operation	The <i>override</i> keyword.
prototype	ActionScript 3: Attribute	The <i>prototype</i> keyword.
rest	ActionScript 3: Parameter	The <i>rest</i> parameter (...).

Topics:

Topic	Detail	See also
Common Conventions	<ul style="list-style-type: none"> Package qualifiers (ActionScript 2) and Packages (ActionScript 3) are generated when the current package is not a namespace root An unspecified type is modeled as <i>var</i> or an empty Type field 	Namespace <small>s_[1495]</small>
ActionScript 3 Conventions	<ul style="list-style-type: none"> The <i>Is Leaf</i> property of a Class corresponds to the sealed keyword If a <i>namespace</i> tag is specified it overrides the <i>Scope</i> that is specified 	

Learn More:

- [Import Source Code](#)^[1510]
- [Generate Source Code](#)^[1490]
- [ActionScript Options](#)^[1530]

11.3.2 Ada 2005

Ada 2005 support is available in the System Engineering and Ultimate editions of Enterprise Architect.

Enterprise Architect supports round trip engineering of Ada 2005, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
adaPackage	Class	A package specification in Ada 2005 without a tagged record.
adaProcedure	Class	A procedure specification in Ada 2005.
delegate	Operation	Access to a subprogram.
enumeration	Inner Class	An <i>enum</i> type.
struct	Inner Class	A record definition.
typedef	Inner Class	A type definition, subtype definition, access type definition, renaming.

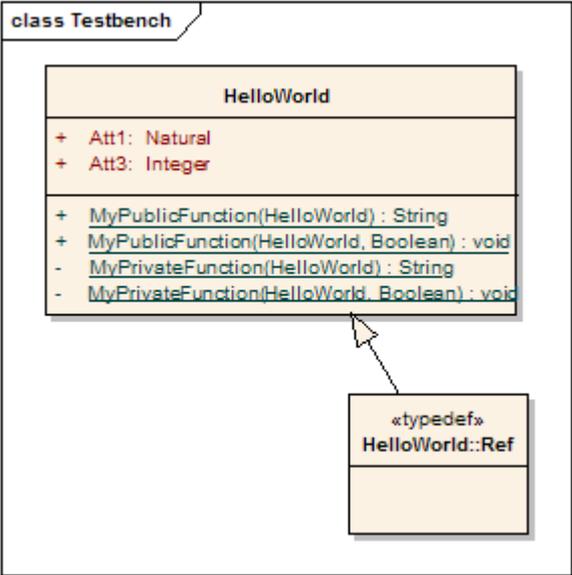
Tagged Values

Reference:

Tag	Applies To	Corresponds To
Discriminant	Inner Class with stereotype <i>typedef</i>	The type's discriminant.
IsAccess	Parameter	Determination of whether the parameter is an access variable.
InstantiatedUnitType	Inner Class with stereotype <i>typedef</i>	The instantiated unit's type (<i>Package / Procedure / Function</i>).
PartType	Inner Class with stereotype <i>typedef</i>	The part type (<i>renames</i> or <i>new</i>).
Type	Inner Class with stereotype <i>typedef</i>	If Value = <i>Sub Type</i> , set subtype. If Value = <i>Access</i> , set access type.

Topics:

Topic	Detail	See also
Other	<ul style="list-style-type: none"> • Appropriate type of source files: Ada specification file, .ads. 	Ada 2005 options

Topic	Detail	See also
Conventions	<ul style="list-style-type: none"> • Ada 2005 imports packages defined as either <<adaPackage>>Class or Class, based on the settings in the Ada 2005 options • A package in the Ada specification file is imported as a Class if it contains a Tagged Record, the name of which is governed by the options Use Class Name for Tagged Record and Alternate Tagged Record Name. All attributes defined in that Tagged Record are absorbed as the Class's attributes • A procedure / function in an Ada specification file is considered as the Class's member function if its first parameter satisfies the conditions specified in the options Ref Param Style, Ignore Reference parameter name and Ref parameter name • The option Define Reference for Tagged Record, if enabled, creates a reference type for the Class, the name of which is determined by the option Reference Type Name <p>For example: HelloWorld.ads</p> <pre> package HelloWorld is type HelloWorld is tagged record Att1: Natural; Att3: Integer; end record; -- Public Functions function MyPublicFunction (P: HelloWorld) return String; procedure MyPublicFunction (P1: in out HelloWorld; AFlag: Boolean); private -- Private Functions function MyPrivateFunction (P: HelloWorld) return String; procedure MyPrivateFunction (P1: in out HelloWorld; AFlag: Boolean); end HelloWorld; </pre>  <pre> classDiagram class HelloWorld { +Att1: Natural +Att3: Integer +MyPublicFunction(HelloWorld): String +MyPublicFunction(HelloWorld, Boolean): void -MyPrivateFunction(HelloWorld): String -MyPrivateFunction(HelloWorld, Boolean): void } class HelloWorldRef { «typedef» } HelloWorldRef --> HelloWorld </pre>	1537

Learn More:

- [Import Source Code](#)^[1510]
- [Generate Source Code](#)^[1490]

11.3.3 C Conventions

Enterprise Architect supports round trip engineering of C, where the following conventions are used:

Stereotype:

Stereotype	Applies To	Corresponds To
enumeration	Inner Class	An <i>enum</i> type
struct	Inner Class	A <i>struct</i> type
	Attribute	A keyword <i>struct</i> in variable definition
typedef	Inner Class	A <i>typedef</i> statement, where the parent is the original type name
union	Inner Class	A <i>union</i> type
	Attribute	A keyword <i>union</i> in variable definition

Tagged Values:

Tag	Applies To	Corresponds To
anonymous	Class also containing the Tagged Value <i>typedef</i>	The name of this class being defined only by the <i>typedef</i> statement
bodyLocation	Operation	The location the method body is generated to; expected values are header , classDec or classBody
typedef	Class with stereotype other than <i>typedef</i>	This Class being defined in a <i>typedef</i> statement

C Code Generation for UML Model:

UML	C Code	Notes
A Class	A pair of C files (.h + .c)	File name is the same as Class name
Operation (public & protected)	Function declaration in .h file and definition in .c file	
Operation (private)	Function definition in .c file only	
Attribute (public & protected)	Variable definition in .h file	
Attribute (private)	Variable definition in .c file	

UML	C Code	Notes
Inner Class (without stereotype)	(N/A)	This inner Class would be ignored

Topics:

Topic	Detail	See also
Capture #define value to be generated in C code	<p>For example, <code>#define PI 3.14</code></p> <ol style="list-style-type: none"> 1. Add an attribute to the Class, with Name = PI and Initial Value = 3.14 2. In the General page of the Attribute Properties dialog, select the Static and Const checkboxes 3. On the Tagged Values page of the Attribute Properties dialog, add a tag called define with the value true 	Attributes ^[695] Attributes Dialog - General ^[696] Attribute Tagged Values ^[699]

Notes:

- Separate conventions apply to **Object Oriented programming in C**

Learn More:

- [Import Source Code](#) ^[1510]
- [Generate Source Code](#) ^[1490]
- [C Options](#) ^[1531]
- [Object Oriented Programming in C](#) ^[1462]

11.3.3.1 Object Oriented Programming In C

The following conventions are used for Object-Oriented programming in C.

To configure Enterprise Architect to support Object-Oriented programming using C, you must set the **Object Oriented Support** option to **True** on the C Specifications page of the Options dialog.

Stereotype

Stereotype	Applies To	Corresponds To
enumeration	Class	An <i>enum</i> type
struct	Class	A <i>struct</i> type
	Attribute	A keyword <i>struct</i> in variable definition
typedef	Class	A <i>typedef</i> s statement, where the parent is the original type name
union	Class	A <i>union</i> type
	Attribute	A keyword <i>union</i> in variable definition

Tagged Values

Tag	Applies To	Corresponds To
anonymous	Class with stereotype of <i>enumeration, struct</i> or <i>union</i>	The name of this Class being defined only by the <i>typedef</i> statement
bodyLocation	Operation	The location the method body is generated to. Expected values are header , classDec or classBody
define	Attribute	<i>#define</i> statement
typedef	Class with stereotype of <i>enumeration, struct</i> or <i>union</i>	This Class being defined in a <i>typedef</i> statement

Topics:

Topic	Detail	See also
Object-Oriented C Code Generation for UML Model	<p>The basic idea of implementing a UML Class in C code is to group the data variable (UML attributes) into a structure type; this structure is defined in a .h file so that it can be shared by other classes and by the client that referred to it</p> <p>An operation in a UML Class is implemented in C code as a function; the name of the function must be a fully qualified name that consists of the operation name, as well as the Class name to indicate that the operation is for that Class</p> <p>A delimiter (specified in the Namespace Delimiter option on the C Specifications page) is used to join the Class name and function (operation) name</p> <p>The function in C code must also have a reference parameter to the Class object - you can modify the Reference as Operation Parameter, Reference Parameter Style and Reference Parameter Name options on the C Specifications page to support this reference parameter</p>	<p>C Specification <small>[1531]</small></p>
Limitations of Object-Oriented Programming in C	<ul style="list-style-type: none"> • No scope mapping for an attribute: an attribute in a UML Class is mapped to a structure variable in C code, and its scope (private, protected or public) is ignored • Currently an inner Class is ignored: if a UML Class is the inner Class of another UML Class, it is ignored when generating C code • Initial value is ignored: the initial value of an attribute in a UML Class is ignored in generated C code 	

Learn More:

- [Import Source Code](#) [1510]
- [Generate Source Code](#) [1490]
- [C Options](#) [1531]

11.3.4 C# Conventions

Enterprise Architect supports the round trip engineering of C#, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
enumeration	Class	An <i>enum</i> type.
event	Operation	An event.
extension	Operation	A Class extension method, represented in code by a <i>this</i> parameter in the signature.
indexer	Operation	A property acting as an index for this Class.
partial	Operation	The <i>partial</i> keyword on an operation.
property	Operation	A property possibly containing both read and write code.
struct	Class	A <i>struct</i> type.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
argumentName	Operation with stereotype <i>extension</i>	The name given to the <i>this</i> parameter.
attributeName	Operation with stereotype <i>property</i> or <i>event</i>	The name of the variable behind this property or event.
className	Operation with stereotype <i>extension</i>	The Class that this method is being added to.
const	Attribute	The <i>const</i> keyword.
definition	Operation with stereotype <i>partial</i>	Whether this is the declaration of the method, or the definition.
delegate	Operation	The <i>delegate</i> keyword.
enumType	Operation with stereotype <i>property</i>	The datatype that the property is represented as.
extern	Operation	The <i>extern</i> keyword.
fixed	Attribute	The <i>fixed</i> keyword.
generic	Operation	The generic parameters for this Operation.
genericConstraints	Templated Class or Interface, Operation with tag <i>generic</i>	The constraints on the generic parameters of this type or operation.
Implements	Operation	The name of the method this implements, including the interface name.
ImplementsExplicit	Operation	The presence of the source interface name in this method declaration.
initializer	Operation	A constructor initialization list.

Tag	Applies To	Corresponds To
new	Class, Interface, Operation	The <i>new</i> keyword.
override	Operation	The <i>override</i> keyword.
params	Parameter	A parameter list using the <i>params</i> keyword.
partial	Class, Interface	The <i>partial</i> keyword.
readonly	Operation with stereotype <i>property</i>	This property only defining read code.
sealed	Operation	The <i>sealed</i> keyword.
static	Class	The <i>static</i> keyword.
unsafe	Class, Interface, Operation	The <i>unsafe</i> keyword.
virtual	Operation	The <i>virtual</i> keyword.
writable	Operation with stereotype <i>property</i>	This property only defining write code.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Namespaces are generated for each package below a namespace root The <i>Const</i> property of an attribute corresponds to the <i>readonly</i> keyword, while the tag <i>const</i> corresponds to the <i>const</i> keyword The value of <i>inout</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ref</i> keyword The value of <i>out</i> for the <i>Kind</i> property of a parameter corresponds to the <i>out</i> keyword Partial Classes can be modeled as two separate Classes with the <i>partial</i> tag The <i>Is Leaf</i> property of a Class corresponds to the <i>sealed</i> keyword 	Namespaces ^[1495]

Learn More:

- [Import Source Code](#) ^[1510]
- [Generate Source Code](#) ^[1490]
- [C# Options](#) ^[1532]

11.3.5 C++ Conventions

Enterprise Architect supports round trip engineering of C++, including the **Managed C++** and **C++/CLI** extensions, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
enumeration	Class	An <i>enum</i> type.
friend	Operation	The <i>friend</i> keyword.
property get	Operation	A read property.
property set	Operation	A write property.
struct	Class	A <i>struct</i> type.
typedef	Class	A <i>typedef</i> statement, where the parent is the original type name.
union	Class	A <i>union</i> type.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
afx_msg	Operation	The <i>afx_msg</i> keyword.
anonymous	Class also containing the Tagged Value <i>typedef</i>	The name of this class being only defined by the <i>typedef</i> statement.
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
bitfield	Attribute	The size, in bits, allowed for storage of this attribute.
bodyLocation	Operation	The location the method body is generated to; expected values are header , classDec or classBody .
callback	Operation	A reference to the CALLBACK macro.
explicit	Operation	The <i>explicit</i> keyword.
initializer	Operation	A constructor initialization list.
inline	Operation	The <i>inline</i> keyword and inline generation of the method body.
mutable	Attribute	The <i>mutable</i> keyword.
throws	Operation	The exceptions that are thrown by this method.
typedef	Class with stereotype other than <i>typedef</i>	This Class being defined in a <i>typedef</i> statement.
typeSynonyms	Class	The <i>typedef</i> name and/or fields of this type.
volatile	Operation	The <i>volatile</i> keyword.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Namespaces are generated for each package below a namespace root <i>By Reference</i> attributes correspond to a pointer to the type specified The <i>Transient</i> property of an attribute corresponds to the <i>volatile</i> keyword The <i>Abstract</i> property of an attribute corresponds to the <i>virtual</i> keyword The <i>Const</i> property of an operation corresponds to the <i>const</i> keyword, specifying a constant return type The <i>Is Query</i> property of an operation corresponds to the <i>const</i> keyword, specifying the method doesn't modify any fields The <i>Pure</i> property of an operation corresponds to a pure <i>virtual</i> method using the "0" syntax The <i>Fixed</i> property of a parameter corresponds to the <i>const</i> keyword 	Namespaces ^[1495]

Learn More:

- [Managed C++ Options](#) ^[1467]
- [C++/CLI Conventions](#) ^[1468]
- [Import Source Code](#) ^[1510]
- [Generate Source Code](#) ^[1490]
- [C++ Options](#) ^[1533]

11.3.5.1 Managed C++ Conventions

The following conventions are used for managed extensions to C++ prior to **C++/CLI**. In order to set Enterprise Architect to generate managed C++ you must modify the C++ version in the **C++ Options**.

Stereotypes

Stereotype	Applies To	Corresponds To
property	Operation	The <code>__property</code> keyword.
property get	Operation	The <code>__property</code> keyword and a read property.
property set	Operation	The <code>__property</code> keyword and a write property.
reference	Class	The <code>__gc</code> keyword.
value	Class	The <code>__value</code> keyword.

Tagged Values

Tag	Applies To	Corresponds To
managedType	Class with stereotype <i>reference</i> , <i>value</i> or <i>enumeration</i> ; Interface	The keyword used in declaration of this type. Expected values are <i>class</i> or <i>struct</i> .

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The <i>typedef</i> and anonymous tags from native C++ are not supported The <i>Pure</i> property of an operation corresponds to the keyword <code>__abstract</code> 	

Learn More:

- [C++/CLI Conventions](#) ^[1468]
- [C++ Options](#) ^[1533]
- [Import Source Code](#) ^[1510]
- [Generate Source Code](#) ^[1490]

11.3.5.2 C++/CLI Conventions

The following conventions are used for modeling C++/CLI extensions to C++. In order to set Enterprise Architect to generate managed C++/CLI you must modify the C++ version in the **C++ Options**.

Stereotypes

Reference:

Stereotype	Applies To	Description
event	Operation	Defines an event to provide access to the event handler for this Class.
property	Operation, Attribute	This is a property possibly containing both read and write code.
reference	Class	Corresponds to the <i>ref class</i> or <i>ref struct</i> keyword.
value	Class	Corresponds to the <i>value class</i> or <i>value struct</i> keyword.

Tagged Values

Reference:

Tag	Applies To	Description
attribute_name	Operation with stereotype <i>property</i> or <i>event</i>	The name of the variable behind this property or event.
generic	Operation	Defines the generic parameters for this Operation.
genericConstraints	Templated Class or Interface, Operation with tag <i>generic</i>	Defines the constraints on the generic parameters for this Operation.
initonly	Attribute	Corresponds to the <i>initonly</i> keyword.
literal	Attribute	Corresponds to the <i>literal</i> keyword.

Tag	Applies To	Description
managedType	Class with stereotype <i>reference</i> , <i>value</i> or <i>enumeration</i> ; Interface	Corresponds to either the <i>class</i> or <i>struct</i> keyword.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The <i>typedef</i> and anonymous tags are not used The <i>property get/property set</i> stereotypes are not used The <i>Pure</i> property of an operation corresponds to the keyword <i>abstract</i> 	

Learn More:

- [C++ Options](#)^[1533]
- [Import Source Code](#)^[1510]
- [Generate Source Code](#)^[1490]

11.3.6 Delphi Conventions

Enterprise Architect supports round trip engineering of Delphi, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
constructor	Operation	A constructor.
destructor	Operation	A destructor.
dispinterface	Class, Interface	A dispatch interface.
enumeration	Class	An enumerated type.
metaclass	Class	A metaclass type.
object	Class	An object type.
operator	Operation	An operator.
property get	Operation	A read property.
property set	Operation	A write property.
struct	Class	A record type.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
overload	Operation	The <i>overload</i> keyword.
override	Operation	The <i>override</i> keyword.
packed	Class	The <i>packed</i> keyword.
property	Class	A property. See Delphi Properties for more information.
reintroduce	Operation	The <i>reintroduce</i> keyword.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The <i>Static</i> property of an attribute or operation corresponds to the <i>class</i> keyword The <i>Fixed</i> property of a parameter corresponds to the <i>const</i> keyword The value of <i>inout</i> for the <i>Kind</i> property of a parameter corresponds to the <i>Var</i> keyword The value of <i>out</i> for the <i>Kind</i> property of a parameter corresponds to the <i>Out</i> keyword 	

Learn More:

- [Import Source Code](#)^[1510]
- [Generate Source Code](#)^[1490]
- [Delphi Options](#)^[1533]
- [Delphi Properties](#)^[1534]

11.3.7 Java Conventions

Enterprise Architect supports round trip engineering of Java - including **AspectJ** extensions - where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
annotation	Interface	An <i>annotation</i> type.
enum	Attributes within a Class stereotyped <i>enumeration</i>	An <i>enumerated</i> option, distinguished from other attributes that have no stereotype.
enumeration	Class	An <i>enum</i> type.
operator	Operation	An operator.
property get	Operation	A read property.

Stereotype	Applies To	Corresponds To
property set	Operation	A write property.
static	Class or Interface	The <i>static</i> keyword.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
annotations	Anything	The annotations on the current code feature.
arguments	Attribute with stereotype <i>enum</i>	The arguments that apply to this enumerated value.
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
dynamic	Class or Interface	The <i>dynamic</i> keyword.
generic	Operation	The generic parameters to this operation.
parameterList	Parameter	A parameter list with the ... syntax.
throws	Operation	The exceptions that are thrown by this method.
transient	Attribute	The transient keyword.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Package statements are generated when the current package is not a namespace root The <i>Const</i> property of an attribute or operation corresponds to the final keyword The <i>Transient</i> property of an attribute corresponds to the volatile keyword The <i>Fixed</i> property of a parameter corresponds to the final keyword 	Namespaces <small>[1495]</small>

Learn More:

- [AspectJ Conventions](#) [1472]
- [Import Source Code](#) [1510]
- [Generate Source Code](#) [1490]
- [Java Options](#) [1535]

11.3.7.1 AspectJ Conventions

The following are the conventions used for supporting AspectJ extensions to Java.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
advice	Operation	A piece of advice in an AspectJ aspect.
aspect	Class	An AspectJ aspect.
pointcut	Operation	A pointcut in an AspectJ aspect.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
className	Attribute or operation within a Class stereotyped <i>aspect</i>	The Classes this AspectJ intertype member belongs to.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The specifications of a pointcut are included in the Behavior field of the method 	

Learn More:

- [Import Source Code](#)^[1510]
- [Generate Source Code](#)^[1490]

11.3.8 PHP Conventions

Enterprise Architect supports the round trip engineering of PHP 4 and 5, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
property get	Operation	A read property.
property set	Operation	A write property.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
attribute_name	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
final	Operations in PHP 5.	The final keyword.

Topics:

Topic	Detail	See also
Common Conventions	<ul style="list-style-type: none"> An unspecified type is modeled as <i>var</i> Methods returning a reference are generated by setting the <i>Return Type</i> to <i>var*</i> Reference parameters are generated from parameters with the parameter <i>Kind</i> set to <i>inout</i> or <i>out</i> 	
PHP 5 Conventions	<ul style="list-style-type: none"> The <i>final</i> Class modifier corresponds to the <i>Is Leaf</i> property The <i>abstract</i> Class modifier corresponds to the <i>Abstract</i> property Parameter type hinting is supported by setting the <i>Type</i> of a parameter The value of <i>inout</i> or <i>out</i> for the <i>Kind</i> property of a parameter corresponds to a <i>reference</i> parameter 	

Learn More:

- [Import Source Code](#)^[1510]
- [Generate Source Code](#)^[1490]
- [PHP Options](#)^[1536]

11.3.9 Python Conventions

Enterprise Architect supports the round trip engineering of Python, where the following conventions are used.

Tagged values

Reference:

Tag	Applies To	Corresponds To
Decorators	Class, Operation	The decorators applied to this element in the source.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Model members with <i>Private Scope</i> correspond to code members with two leading underscores Attributes are only generated when the Initial value is not empty 	

Topic	Detail	See also
	<ul style="list-style-type: none"> All types are reverse engineered as <i>var</i> 	

Learn More:

- [Import Source Code](#) ^[1510]
- [Generate Source Code](#) ^[1490]
- [Python Options](#) ^[1536]

11.3.10 System C Conventions

Enterprise Architect supports round-trip engineering of SystemC, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
delegate	Method	A delegate.
enumeration	Inner Class	An <i>enum</i> type.
friend	Method	A <i>friend</i> method.
property	Method	A property definition.
sc_ctor	Method	A SystemC constructor.
sc_module	Class	A SystemC module.
sc_port	Attribute	A port.
sc_signal	Attribute	A signal
struct	Inner Class	A <i>struct</i> or <i>union</i> .

Tagged Values

Reference:

Tag	Applies To	Corresponds To
kind	Attribute (Port)	Port kind (<i>clocked, fifo, master, slave, resolved, vector</i>).
mode	Attribute (Port)	Port mode (<i>in, out, inout</i>).
overrides	Method	The <i>Inheritance</i> list of a method declaration.
throw	Method	The exception specification of a method.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> SystemC also inherits most of the stereotypes and Tagged Values of C++ 	C++ Conventions <small>[1465]</small>

SystemC Toolbox Pages

To access the **SystemC** pages of the **Toolbox**, select the **More tools | HDL | SystemC Constructs** menu option. Drag these icons onto a diagram to model a SystemC design.

Reference:

Page	Item	Action
SystemC	Module	Define a SystemC Module. <i>An <code>sc_module</code>-stereotyped Class element.</i>
	Enumeration	Define an Enumerated Type. <i>An <code>enumeration</code>-stereotyped Enumeration element.</i>
	Struct	Define a Structure. <i>A <code>struct</code>-stereotyped Class element.</i>
SystemC Features	Port	Define a SystemC Port. <i>An <code>sc_port</code>-stereotyped attribute.</i>
	Signal	Define a SystemC Signal. <i>An <code>sc_signal</code>-stereotyped attribute.</i>
	Constructor	Define a SystemC Constructor. <i>An <code>sc_ctor</code>-stereotyped method.</i>

Learn More:

- [Import Source Code](#) [1510]
- [Generate Source Code](#) [1490]
- [SystemC Options](#) [1537]

11.3.11 VB.Net Conventions

Enterprise Architect supports round-trip engineering of Visual Basic.Net, where the following conventions are used. Earlier versions of **Visual Basic** are supported as a different language.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
event	Operation	An event declaration.

Stereotype	Applies To	Corresponds To
import	Operation	An operation to be imported from another library.
module	Class	A module.
operator	Operation	An operator overload definition.
partial	Operation	The <i>partial</i> keyword on an operation.
property	Operation	A property possibly containing both read and write code.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
Alias	Operation with stereotype <i>import</i>	The alias for this imported operation.
attribute_name	Operation with stereotype <i>property</i>	The name of the variable behind this property.
Charset	Operation with stereotype <i>import</i>	The <i>character set</i> clause for this import. One of the values <i>Ansi</i> , <i>Unicode</i> or <i>Auto</i> .
delegate	Operation	The <i>Delegate</i> keyword.
enumTag	Operation with stereotype <i>property</i>	The datatype that this property is represented as.
Handles	Operation	The <i>handles</i> clause on this operation.
Implements	Operation	The <i>implements</i> clause on this operation.
Lib	Operation with stereotype <i>import</i>	The library this import comes from.
MustOverride	Operation	The <i>MustOverride</i> keyword.
Narrowing	Operation with stereotype <i>operator</i>	The <i>Narrowing</i> keyword.
NotOverrideable	Operation	The <i>NotOverrideable</i> keyword.
Overloads	Operation	The <i>Overloads</i> keyword.
Overrides	Operation	The <i>Overrides</i> keyword.
parameterArray	Parameter	A parameter list using the <i>ParamArray</i> keyword.
partial	Class, Interface	The <i>Partial</i> keyword.
readonly	Operation with stereotype <i>property</i>	This property only defining read code.
shadows	Class, Interface, Operation	The <i>Shadows</i> keyword.
Shared	Attribute	The <i>Shared</i> keyword.
Widening	Operation with stereotype <i>operator</i>	The <i>Widening</i> keyword.
writeonly	Operation with stereotype <i>property</i>	This property only defining write code.

Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> Namespaces are generated for each package below a namespace root The <i>Is Leaf</i> property of a Class corresponds to the <i>NotInheritable</i> keyword The <i>Abstract</i> property of a Class corresponds to the <i>MustInherit</i> keyword The <i>Static</i> property of an attribute or operation corresponds to the <i>Shared</i> keyword The <i>Abstract</i> property of an operation corresponds to the <i>MustOverride</i> keyword The value of <i>in</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ByVal</i> keyword The value of <i>inout</i> or <i>out</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ByRef</i> keyword 	Namespaces ^[1495]

Learn More:

- [Visual Basic Conventions](#) ^[1480]
- [Import Source Code](#) ^[1510]
- [Generate Source Code](#) ^[1490]
- [VB.Net Options](#) ^[1537]

11.3.12 Verilog Conventions

Enterprise Architect supports round-trip engineering of Verilog, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
asynchronous	Method	A concurrent process.
enumeration	Inner Class	An <i>enum</i> type.
initializer	Method	An initializer process.
module	Class	A module.
part	Attribute	A component instantiation.
port	Attribute	A port.
synchronous	Method	A sequential process.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
kind	Attribute (signal)	The signal kind (such as <i>register</i> , <i>bus</i>).
mode	Attribute (port)	The port mode (<i>in</i> , <i>out</i> , <i>inout</i>).
Portmap	Attribute (part)	The generic / port map of the component instantiated.
sensitivity	Method	The sensitivity list of a sequential process.
type	Attribute	The range or type value of an attribute.

Verilog Toolbox Pages

To access the **Verilog** pages of the **Toolbox**, select the **More tools | HDL | Verilog Constructs** menu option. Drag these icons onto a diagram to model a Verilog design.

Reference:

Page	Item	Action
Verilog	Module	Define a Verilog Module. <i>A module-stereotyped Class element.</i>
	Enumeration	Define an Enumerated Type. <i>An enumeration-stereotyped Class element.</i>
Verilog Features	Port	Define a Verilog Port. <i>A port-stereotyped attribute.</i>
	Part	Define a Verilog component instantiation <i>A part-stereotyped attribute.</i>
	Attribute	Define an attribute.
	Procedure <ul style="list-style-type: none"> • Concurrent • Sequential • Initializer. 	Define a Verilog process: <ul style="list-style-type: none"> • An <i>asynchronous</i>-stereotyped method • A <i>synchronous</i>-stereotyped method • An <i>initializer</i>-stereotyped method

Learn More:

- [Import Source Code](#) [1510]
- [Generate Source Code](#) [1490]
- [Verilog Options](#) [1538]

11.3.13 VHDL Conventions

Enterprise Architect supports round-trip engineering of VHDL, where the following conventions are used.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
architecture	Class	An architecture.
asynchronous	Method	An asynchronous process.
configuration	Method	A configuration.
enumeration	Inner Class	An <i>enum</i> type.
entity	Interface	An entity.
part	Attribute	A component instantiation.
port	Attribute	A port.
signal	Attribute	A signal declaration.
struct	Inner Class	A record definition.
synchronous	Method	A synchronous process.
typedef	Inner Class	A <i>type</i> or <i>sub type</i> definition.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
isGeneric	Attribute (port)	The port declaration in a generic interface.
isSubType	Inner Class (typedef)	A subtype definition.
kind	Attribute (signal)	The signal kind (such as <i>register</i> , <i>bus</i>).
mode	Attribute (port)	The port mode (<i>in</i> , <i>out</i> , <i>inout</i> , <i>buffer</i> , <i>linkage</i>).
portmap	Attribute (part)	The generic / port map of the component instantiated.
sensitivity	Method (synchronous)	The sensitivity list of a synchronous process.
type	Inner Class (typedef)	The type indication of a type declaration.
typeNameSpace	Attribute (part)	The type namespace of the instantiated component.

VHDL Toolbox Pages

To access the **VHDL** pages of the **Toolbox**, select the **More tools | HDL | VHDL Constructs** menu option. Drag these icons onto a diagram to model a VHDL design.

Reference:

Page	Item	Action
VHDL	Architecture	Define an architecture to be associated with a VHDL entity. <i>An architecture-stereotyped Class element.</i>
	Entity	Define a VHDL entity to contain the Port definitions. <i>An entity-stereotyped interface element.</i>
	Enumeration	Define an Enumerated Type. <i>An enumeration-stereotyped enumeration element.</i>
	Struct	Define a VHDL record. <i>A struct-stereotyped Class element.</i>
	Typedef	Define a VHDL type or subtype <i>A typedef-stereotyped Class element.</i>
VHDL Features	Port	Define a VHDL Port. <i>A port-stereotyped attribute.</i>
	Part	Define a VHDL component instantiation <i>A part-stereotyped attribute.</i>
	Signal	Define a VHDL signal. <i>A signal-stereotyped attribute.</i>
	Procedure <ul style="list-style-type: none"> • Concurrent • Sequential • Configuration. 	Define a VHDL process: <ul style="list-style-type: none"> • <i>An asynchronous-stereotyped method</i> • <i>A synchronous-stereotyped method</i> • <i>A configuration-stereotyped method</i>

Learn More:

- [Import Source Code](#)^[1510]
- [Generate Source Code](#)^[1490]
- [VHDL Options.](#)^[1538]

11.3.14 Visual Basic Conventions

Enterprise Architect supports the round trip engineering of Visual Basic 5 and 6, where the following conventions are used. **Visual Basic .Net** is supported as a different language.

Stereotypes

Reference:

Stereotype	Applies To	Corresponds To
global	Attribute	The <i>Global</i> keyword.
import	Operation	An operation to be imported from another library.

Stereotype	Applies To	Corresponds To
property get	Operation	A property get.
property set	Operation	A property set.
property let	Operation	A property let.
with events	Attribute	The <i>WithEvents</i> keyword.

Tagged Values

Reference:

Tag	Applies To	Corresponds To
Alias	Operation with stereotype <i>import</i>	The alias for this imported operation.
attribute_name	Operation with stereotype <i>property get</i> , <i>property set</i> or <i>property let</i>	The name of the variable behind this property.
Lib	Operation with stereotype <i>import</i>	The library this import comes from.
New	Attribute	The <i>New</i> keyword.

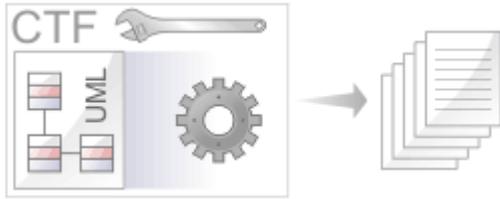
Topics:

Topic	Detail	See also
Other Conventions	<ul style="list-style-type: none"> The value of <i>in</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ByVal</i> keyword The value of <i>inout</i> or <i>out</i> for the <i>Kind</i> property of a parameter corresponds to the <i>ByRef</i> keyword 	

Learn More:

- [Visual Basic .Net Conventions](#) ^[1475]
- [Import Source Code](#) ^[1510]
- [Generate Source Code](#) ^[1490]
- [Visual Basic Options](#) ^[1539]

11.4 Code Template Framework



The Code Template Framework (CTF) is used during forward engineering UML models. The CTF enables you to:

- Generate source code from UML models
- Customize the way in which Enterprise Architect generates source code
- Forward engineer languages not specifically supported by Enterprise Architect

Topics:

Topic	Detail	See also
Default Templates	Default Code Templates which are built into Enterprise Architect for forward engineering supported languages	Code Templates ^[1482]
Code Template Editor	A Code Template Editor for creating and maintaining user-defined Code Templates	Code Template Editor ^[1485]
Synchronize Code	Code Templates to synchronize code.	Synchronize code ^[1486]

11.4.1 Code Templates

Topics:

Topic	Detail	See also
Usage	<p>Code templates enable you to customize code generation of existing languages. For example:</p> <ul style="list-style-type: none"> • Modify the file headers created when generating new files • Change the style of the generated code (such as indenting or brace position) to match the required coding standards • Handle particular stereotypes to generate things like specialized method bodies and extra methods <p>They also enable you to add code generation of entirely new languages that Enterprise Architect would otherwise not be able to handle. In this situation it is most useful to combine code templates with an MDG technology file that includes the datatypes, and options for default file extensions.</p> <p>Enterprise Architect's base code templates specify the transformation from UML elements to the various parts of a given programming language. The templates are written as plain text with a syntax that shares some aspects of both</p>	Add Code Modules ^[1070] Base Templates ^[1483]

	<p>mark-up languages and scripting languages. A simple example of a template used by Enterprise Architect is the 'Class template'. It is used to generate source code from a UML Class:</p> <pre>%ClassNotes% %ClassDeclaration% %ClassBody%</pre> <p>The above template simply refers to three other templates, namely <i>ClassNotes</i>, <i>ClassDeclaration</i> and <i>ClassBody</i>. The enclosing percent (%) signs indicate a macro. Code Templates consist of various types of macros, each resulting in a substitution in the generated output. For a language such as C++, the result of processing the above template might be:</p> <pre>/** * This is an example class note * generated using code templates * @author Sparx Systems */ class ClassA: public ClassB { ... }</pre>	
Execution of Code Templates	<p>A reference to a template (such as the <i>%ClassNotes%</i> macro, from our example above) results in the execution of that template.</p> <p>Each template is designed for use with a particular element. For example the <i>ClassNotes</i> template is to be used with UML Class elements.</p> <p>The element that is currently being generated is said to be <i>in scope</i>. If the element in scope is stereotyped Enterprise Architect looks for a template that has been defined for that stereotype. If a match is found, the specialized template is executed. Otherwise the default implementation of the base template is used.</p> <p>Templates are processed sequentially, line by line, replacing each macro with its underlying text value from the model.</p>	

11.4.1.1 Base Templates

The Code Template Framework consists of a number of base templates. Each base template transforms particular aspects of the UML to corresponding parts of object-oriented languages.

The following table lists and briefly describes the base templates used in the CTF.

Template	Description
Attribute	A top-level template to generate member variables from UML attributes
Attribute Declaration	Used by the <i>Attribute</i> template to generate a member variable declaration
Attribute Notes	Used by the <i>Attribute</i> template to generate member variable notes
Class	A top-level template for generating Classes from UML Classes
Class Base	Used by the <i>Class</i> template to generate a base Class name in the inheritance list of a derived Class, where the base Class doesn't exist in the model

Template	Description
Class Body	Used by the <i>Class</i> template to generate the body of a Class
Class Declaration	Used by the <i>Class</i> template to generate the declaration of a Class
Class Interface	Used by the <i>Class</i> template to generate an interface name in the inheritance list of a derived Class, where the interface doesn't exist in the model
Class Notes	Used by the <i>Class</i> template to generate the Class notes
File	A top-level template for generating the source file For languages such as C++, this corresponds to the header file
Import Section	Used in the <i>File</i> template to generate external dependencies
Linked Attribute	A top-level template for generating attributes derived from UML Associations
Linked Attribute Notes	Used by the <i>Linked Attribute</i> template to generate the attribute notes
Linked Attribute Declaration	Used by the <i>Linked Attribute</i> template to generate the attribute declaration
Linked Class Base	Used by the <i>Class</i> template to generate a base Class name in the inheritance list of a derived Class, for a Class element in the model that is a parent of the current Class
Linked Class Interface	Used by the <i>Class</i> template to generate an Interface name in the inheritance list of a derived Class, for an Interface element in the model that is a parent of the current Class
Namespace	A top-level template for generating namespaces from UML packages (although not all languages have namespaces, this template can be used to generate an equivalent construct, such as packages in Java)
Namespace Body	Used by the <i>Namespace</i> template to generate the body of a namespace
Namespace Declaration	Used by the <i>Namespace</i> template to generate the namespace declaration
Operation	A top-level template for generating operations from a UML Class's operations
Operation Body	Used by the <i>Operation</i> template to generate the body of a UML operation
Operation Declaration	Used by the <i>Operation</i> template to generate the operation declaration
Operation Notes	Used by the <i>Operation</i> template to generate documentation for an operation
Parameter	Used by the <i>Operation Declaration</i> template to generate parameters

The second table lists templates used for generating code for languages that have separate interface and implementation sections.

Template	Description
Class Impl	A top-level template for generating the implementation of a Class
Class Body Impl	Used by the <i>Class Impl</i> template to generate the implementation of Class members
File Impl	A top-level template for generating the implementation file

Template	Description
File Notes Impl	Used by the <i>File Impl</i> template to generate notes in the source file
Import Section Impl	Used by the <i>File Impl</i> template to generate external dependencies
Operation Impl	A top-level template for generating operations from a UML Class's operations
Operation Body Impl	Used by the <i>Operation</i> template to generate the body of a UML operation
Operation Declaration Impl	Used by the <i>Operation</i> template to generate the operation declaration
Operation Notes Impl	Used by the <i>Operation</i> template to generate documentation for an operation

Topics:

Topic	Detail	See also
Further Information	<p>The base templates form a hierarchy, which varies slightly across different programming languages. A typical template hierarchy relevant to a language like C# or Java (which do not have header files) is shown in the example diagram below. In this diagram the templates are modeled as Classes (in reality they are just plain text). This hierarchy would be slightly more complicated for languages like C++ and Delphi, which have separate implementation templates.</p> <p>Each of the base templates must be specialized to be of use in code engineering; in particular, each template is specialized for the supported languages (or 'products')</p> <p>For example, there is a <i>ClassBody</i> template defined for C++, another for C#, another for Java, and so on; by specializing the templates, you can tailor the code generated for the corresponding UML entity</p> <p>Once the base templates are specialized for a given language, they can be further specialized based on:</p> <ul style="list-style-type: none"> • A Class's stereotype • A feature's stereotype (where the feature can be an operation or attribute) <p>This type of specialization enables, for example, a C# operation that is stereotyped as «property» to have a different <i>Operation Body</i> template from an ordinary operation; the <i>Operation Body</i> template can then be specialized further, based on the Class stereotype</p>	

11.4.2 The Code Template Editor

The Code Template Editor provides the facilities of the *Common Code Editor*, including intellisense for the various macros. For more information on intellisense and the Common Code Editor, see the **Code Editors** topic.

Access: [Settings | Code Generation Templates](#)

Reference:

Option	Action	See also
Language	Select the programming language	
New Language	Display the Programming Languages Datatypes dialog, which enables you to include programming languages other than those supported for Enterprise Architect, for which to create or edit code templates	Programming Languages Datatypes ^[777]
Template	Display the contents of the active template, and provides the editor for modifying templates	
Templates	List the base code templates; the active template is highlighted The Modified field indicates whether you have changed the default template for the current language	
Stereotype Overrides	List the stereotyped templates, for the active base template The Modified field indicates whether you have modified a default stereotyped template	
Add New Custom Template	Invoke a dialog for creating a custom stereotyped template	
Add New Stereotyped Override	Invoke a dialog for adding a stereotyped template, for the currently selected base template	
Get Default Template	Update the editor display with the default version of the active template	
Save	Overwrite the active templates with the contents of the editor	
Delete	If you have overridden the active template, the override is deleted and replaced by the corresponding default code template	

Notes:

- User-modified and user-defined Code Templates can be imported and exported as Reference Data (see the Sharing Reference Data topic. The templates defined for each language are indicated in the Export Reference Data dialog by the language name with the suffix `_Code_Templates`. If no templates exist for a language, there is no entry for the language in the dialog

Learn More:

- [The Code Template Editor in MDG Development](#)^[1147]
- [Sharing Reference Data](#)^[236]
- [Code Editors](#)^[1394]

11.4.3 Synchronize Code

Enterprise Architect uses code templates during the forward synchronization of the following programming languages:

- ActionScript
- C
- C++
- C#

- Delphi
- Java
- PHP
- Python
- VB
- VB.Net

Only a subset of the code templates are used during synchronization. This subset corresponds to the distinct sections that Enterprise Architect recognizes in the source code. The following table lists the code templates and their corresponding code sections, which can be synchronized.

Code Template	Code Section
Class Notes	Comments preceding Class declaration.
Class Declaration	Up to and including Class parents.
Attribute Notes	Comments preceding Attribute declaration.
Attribute Declaration	Up to and including terminating character.
Operation Notes	Comments preceding operation declaration.
Operation Notes Impl	As for <i>Operation Notes</i> .
Operation Declaration	Up to and including terminating character.
Operation Declaration Impl	Up to and including terminating character.
Operation Body	Everything between and including the braces.
Operation Body Impl	As for <i>Operation Body</i> .

Topics:

Topic	Detail	See also
Change Types	<p>Three types of change can occur in the source when it is synchronized with the UML model:</p> <ul style="list-style-type: none"> • Synchronize Existing Sections: for example, changing the return type in an operation declaration • Add New Sections to Existing Features: for example, adding notes to a Class declaration, where there were previously none • Add New Features and Elements: for example, adding a new operation to a Class <p>Each of these changes must be handled differently by Enterprise Architect; their effect on the CTF is described in the linked topics above.</p>	<p>Synchronize Existing Sections ^[1487]</p> <p>Add New Sections to Existing Features ^[1488]</p> <p>Add New Features and Elements ^[1488]</p>

11.4.3.1 Synchronize Existing Sections

Topics:

Topic	Detail	See also
Usage	When an existing section in the source code differs from the	

	<p>result generated by the corresponding template, that section is replaced. Consider for example, the following C++ Class declaration:</p> <pre>(asm) class A: public B</pre> <p>Now assume you add an inheritance relationship from Class A to Class C; the entire Class declaration would be replaced with something like:</p> <pre>(asm) class A: public B, public C</pre>	
--	--	--

11.4.3.2 Add New Sections

Topics:

Topic	Detail	See also
Usage	<p>The following can be added as new sections, to existing features in the source code:</p> <ul style="list-style-type: none"> • Class Notes • Attribute Notes • Operation Notes • Operation Notes Impl • Operation Body • Operation Body Impl <p>Assume Class A from the previous example had no note when you originally generated the code. Now assume that you specify a note in the model for Class A. Enterprise Architect attempts to add the new note from the model during synchronization. It does this by executing the <i>Class Notes</i> template.</p> <p>To make room for the new section to be inserted, you can specify how much white space to append to the section via synchronization macros. These macros are described in the Control Macros topic.</p>	Control Macros <small>[1136]</small>

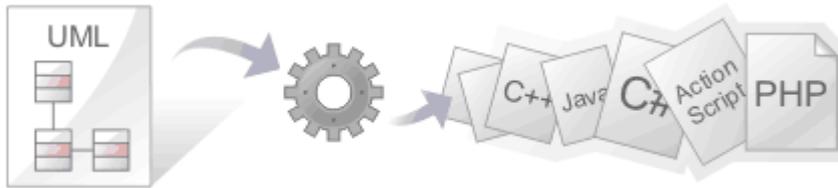
11.4.3.3 Add New Features and Elements

Topics:

Topic	Detail	See also
Usage	<p>The following features and elements can be added to the source code during synchronization:</p> <ul style="list-style-type: none"> • Attributes • Inner Classes • Operations <p>These are added by executing the relevant templates for each new element or feature in the model. Enterprise Architect attempts to preserve the appropriate indenting of new features in the code, by finding the indents specified in list macros of the Class. For languages that make use of namespaces, the <i>synchNamespaceBodyIndent</i> macro is available. Classes defined within a (non-global) namespace are indented</p>	

	<p>according to the value set for this macro, during synchronization. This value is ignored for Classes defined within a package setup as a root namespace, or if the Generate Namespace option is set to False in the appropriate language page (C#, C++ or VB.Net) on the Options dialog (Tools Options Source Code Engineering <language>).</p>	
--	--	--

11.5 Generate Source Code



Generating source code (forward engineering) takes the UML Class or Interface model elements and creates a source code equivalent for future elaboration and compilation. By forward engineering code from the model, the mundane work involved with having to key in Classes and attributes and methods is avoided, and symmetry between model and code is ensured.

Topics:

Topic	Detail	See also
Languages	Enterprise Architect supports code generation in each of the following languages: <ul style="list-style-type: none"> • Action Script • C • C# (for both .NET 1.1 and .NET 2.0) • C++ (standard, plus .NET managed C++ extensions) • Delphi • Java (including Java 1.5, Aspects and Generics) • PHP • Python • Visual Basic • Visual Basic .NET 	
Elements	Code is generated from Class or Interface model elements, so you must create the required Class and Interface elements to generate from Add attributes (which become variables) and operations (which become methods)	
Settings	Before you generate code, you should ensure the default settings for code generation match your requirements; set up the defaults to match your required language and preferences Preferences that you can define include default constructors and destructors, methods for interfaces and the Unicode options for created languages Languages such as Java support 'namespaces' and can be configured to specify a namespace root In addition to the default settings for generating code, Enterprise Architect facilitates setting specific generation options for each of the supported languages	Source Code Options ^[1517] Namespaces ^[1495] ActionScript Options ^[1458] C Options ^[1461] C# Options ^[1463] C++ Options ^[1465] Delphi Options ^[1469] Java Options ^[1470] PHP Options ^[1472] Python Options ^[1473] Visual Basic Options ^[1480] Visual Basic .NET Options ^[1475]

Code Template Framework	The Code Template Framework (CTF) enables you to customize the way Enterprise Architect generates source code and also enables generation of languages that are not specifically supported by Enterprise Architect	Code Template Framework (CTF) ^[1482]
Local Paths	Local path names enable you to substitute tags for directory names	Local Paths ^[1523] Local Paths Dialog ^[1524]
Behavioral Code	You can also generate code from three UML behavioral modeling paradigms: <ul style="list-style-type: none"> • State Machine diagrams (SW & HW) • Interaction (Sequence) diagrams (SW) • Activity diagrams (SW) 	State Machine diagrams ^[1497] Interaction (Sequence) diagrams ^[1503] Activity diagrams ^[1504]
Live Code Generation	On the Code Engineering submenu, you have the option to update your source code instantly as you make changes to your model	Code Engineering Sub-Menu ^[447]
Tasks	When you generate code, you perform one or more of the following tasks: <ul style="list-style-type: none"> • Generate a Single Class • Generate a Group of Classes • Generate a Package • Update Package Contents 	Generate a Single Class ^[1491] Generate a Group of Classes ^[1492] Generate a Package ^[1493] Update Package Contents ^[1494]

Notes:

- The tools provided by Enterprise Architect for code engineering and debugging are available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- When security is enabled there are permissions for Generate Source Code and DDL and Reverse Engineer from DDL and Source Code

Learn More:

- [Importing Source Code](#) ^[1508]
- [Generate From Behavioral Models](#) ^[1496]

11.5.1 Generate a Single Class

To generate code for a single Class, first ensure the design of the model element (Class or Interface) is complete. Also ensure you have added Inheritance connectors to parents and associations to other Classes that are used. Also add Inheritance connectors to Interfaces that your Class implements; Enterprise Architect offers the option to generate function stubs for all interface methods that a Class implements. Once the design is satisfactory, follow the procedure below.

To Generate Code for a Single Class, follow the steps below

How To:

Step	Action	See also
1	Open the diagram containing the Class or Interface for which to generate code.	

Step	Action	See also
2	Right-click on the required Class or Interface to display the context menu and select the Generate Code menu option, or press (F11). The Generate Code dialog displays, which enables you to control how and where your source code is generated.	
3	On the Path field, click on (...) (Browse) and select a path name for your source code to be generated to.	
4	In the Target Language field, click on the drop-down arrow and select the language to generate; this becomes the permanent option for that Class, so change it back if you are only doing one pass in another language.	
5	Click on the Advanced button. The Object Options dialog displays.	
6	Set any custom options (for this Class alone), then click on the Close button to return to the Generate Code dialog.	
7	In the Import(s) / Header(s) fields, type any import statements, <i>#includes</i> or other header information. (Note that in the case of Visual Basic this information is ignored; in the case of Java the two import text boxes are merged; and in the case of C++ the first import text area is placed in the header file and the second in the body (.cpp) file.)	
8	Click on the Generate button to create the source code.	
9	When complete, click on the View button to see what has been generated. Note that you should set up your default viewer/editor for each language type first. You can also set up the default editor on the Code Editors page of the Options dialog (Tools Options Source Code Engineering Code Editors).	The Source Code Viewer ^[1408]

11.5.2 Generate a Group of Classes

In addition to being able to generate code for an individual Class, you can also select a group of Classes for batch code generation. When you do this, you accept all the default code generation options for each Class in the set.

Steps:

Step	Detail	See also
1	Select a group of Classes and/or interfaces in a diagram	
2	Right-click on an element in the group to display the context menu.	
3	Select the Code Generation Generate Selected elements menu option. The Save As dialog displays, on which you specify the file path and name for each code file. Enter this information and click on the Save button	
4	The Batch Generation dialog displays, showing the status of the process as it executes (the process might be too fast to see this dialog).	

Notes:

- If any of the elements selected are not Classes or interfaces the option to generate code is not available

Learn More:

- [Synchronize Model and Code](#)^[1506]

11.5.3 Generate a Package

In addition to generating source code from single Classes and groups of Classes, you can also generate code from a package. This feature provides options to recursively generate child packages and automatically generate directory structures based on the package hierarchy. This enables you to generate a whole branch of your project model in one step.

To generate a package, follow the steps below

How To:

Step	Action	See also
1	In the Project Browser , right-click on the package to generate code for. The context menu displays.	
2	Select the Code Engineering Generate Source Code menu option. The Generate Package Source Code dialog displays.	
3	In the Synchronize field, click on the drop-down arrow and select the appropriate synchronize option: <ul style="list-style-type: none"> • Synchronize model and code: Classes with existing files are forward synchronized with that file; Classes with no existing file are generated to the displayed target file • Overwrite code: All selected target files are overwritten (forward generated) • Do not generate: Only selected Classes that do not have an existing file are generated; all other Classes are ignored 	
4	Highlight the Classes to generate. Leave unselected any to not generate. If you want to display the information in a more readable layout, you can resize the dialog and its columns.	
5	To make Enterprise Architect automatically generate directories and filenames based on the package hierarchy, select the Auto Generate Files checkbox. This then enables the Root Directory field, in which you select a root directory under which the source directories are to be generated. By default, the Auto Generate Files feature <i>ignores</i> any file paths that are already associated with a Class. You can change this behavior by also selecting the Retain Existing File Paths checkbox.	
6	To include all sub-packages in the output, select the Include Child Packages checkbox.	
7	Click on the Generate button to start generating code.	

As code generation proceeds Enterprise Architect displays progress messages. If a Class requires an output filename Enterprise Architect prompts you to enter one at the appropriate time (assuming **Auto Generate Files** is not selected). For example, if the selected Classes include partial Classes, a prompt displays to enter the filename for the second partial Class.

For additional information on the options on the **Generate Package Source Code** dialog, see the following

table:

Option	Action	See also
Root Package	Check the name of the package to be generated.	
Synchronize	Select options that specify how existing files should be generated.	
Auto Generate Files	Specify whether Enterprise Architect should automatically generate file names and directories, based on the package hierarchy.	
Root Directory	If Auto Generate Files is selected, display the path under which the generated directory structures are created.	
Retain Existing File Paths	If Auto Generate Files is selected, specify whether to use existing file paths associated with Classes. If unselected, Enterprise Architect generates Classes to automatically determined paths, regardless of whether source files are already associated with Classes.	
Include all Child Packages	Include all Classes from all sub-packages of the target package in the list. This option facilitates recursive generation of a given package and its sub-packages.	
Select Objects to Generate	List all Classes that are available for generation under the target packages. Only selected (highlighted) Classes are generated. Classes are listed with their target source file.	
Select All	Mark all Classes in the list as selected.	
Select None	Mark all Classes in the list as unselected.	
Generate	Start the generation of all selected Classes.	
Cancel	Exit the Generate Package Source Code dialog. No Classes are generated.	

11.5.4 Update Package Contents

To synchronize a directory tree, follow the steps below

How To:

Step	Action	See also
1	In the Project Browser , right-click on the root package of the tree to synchronize. The context menu displays.	
2	Select the Code Engineering Synchronize Package With Code menu option. The Synchronize Package Contents dialog displays.	
3	In the Update Type panel, select the radio button to Forward Engineer or Reverse Engineer the package Classes.	
4	To include child packages in the synchronization, select the Include child packages in generation checkbox.	
5	Click on the OK button to start.	

Enterprise Architect uses the directory names specified when the project source was first imported/generated and updates either the model or the source code depending on the option chosen.

11.5.5 Namespaces

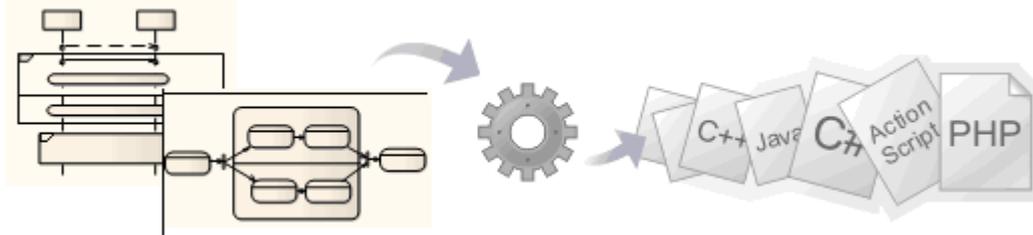
Topics:

Topic	Detail	See also
Usage	<p>Languages such as Java support package structures or namespaces. Enterprise Architect lets you specify a package as a namespace root, which denotes where the namespace structure starts; all subordinate packages below this point are generated as namespaces to code.</p> <p>To define a package as a namespace root, right-click on the package in the Project Browser and select the Code Engineering Set as Namespace Root context menu option. The package icon in the Project Browser changes to include a colored corner ().</p> <p>When you have set the namespace root, the menu option changes to Clear Namespace Root; click on this option to take the namespace root status off the package. (Also, see the context menu described below.)</p> <p>Once you have set a namespace root, Java code generated beneath this root automatically adds a package declaration at the head of the generated file indicating the current package location.</p> <p>To view a list of namespaces, select the Settings Namespace Roots menu option. The Namespaces dialog displays.</p> <p>If you double-click on a namespace in the list, the package is highlighted in the Project Browser. Alternatively, right-click on the namespace to display a context menu, and select the Locate Package in Browser menu option.</p> <p>You can also clear the selected namespace, by selecting the Clear Namespace Attribute option.</p>	

Notes:

- When performing code generation, any package name that contains whitespace characters is automatically treated as a namespace root

11.6 Generate From Behavioral Models



Enterprise Architect's system engineering capability facilitates code generation from each of the following UML behavioral diagrams:

Topic	Link
State Machine diagrams (SW & HW)	State Machine diagrams ^[1497]
Interaction (Sequence) diagrams (SW)	Interaction (Sequence) diagrams ^[1503]
Activity diagrams (SW)	Activity diagrams ^[1504]

You can generate code in various software and **hardware** languages, including C(OO), C++, C#, Java, VB, Net, VHDL, Verilog and SystemC.

To experiment with code generation from these diagrams, using the EAExample project provided with your Enterprise Architect installer, follow the steps below

How To:

Step	Action	See also
1	Open the <i>EAExample.eap</i> file by selecting the Help Open Example Model menu option.	
2	<p>From the Project Browser, select any of the following packages:</p> <p>Software Language Examples:</p> <ul style="list-style-type: none"> Project Models > System Model > Implementation Model (PSM) > Java Model With Behaviors. Generate the Account and Order classes Project Models > Systems Engineering Model > Implementation Model > Software > C#. Generate the DataProcessor Class Project Models > Systems Engineering Model > Implementation Model > Software > C++. Generate the IO Class Project Models > Systems Engineering Model > Implementation Model > Software > Java. Generate the IO Class Project Models > Systems Engineering Model > Implementation Model > Software > VBNet. Generate the IO Class <p>Hardware Language Examples:</p> <ul style="list-style-type: none"> Project Models > Systems Engineering Model > Implementation Model > Hardware > SystemC. Generate the PlayBack Class Project Models > Systems Engineering Model > Implementation Model > Hardware > VHDL. Generate the PlayBack Class 	

Step	Action	See also
	<ul style="list-style-type: none"> Project Models > Systems Engineering Model > Implementation Model > Hardware > Verilog. Generate the PlayBack Class 	
3	When completed, press (Ctrl+E) to open the generated source code. You should see methods generated in the code.	

Notes:

- Software code generation from behavioral models is available in the Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- Hardware code generation from State Machine models is available in the Systems Engineering and Ultimate editions of Enterprise Architect
- For C(OO), please ensure that, on the C Specifications page of the Options dialog, you have set the Object Oriented Support option to True
- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class

Learn More:

- [State Machine modeling for HDLs](#) ^[1502]

11.6.1 SW Code Generation - State Machine Diagrams

A State Machine in a *Class* internally generates the following constructs in software languages to enable effective execution of the States' behaviors (**do**, **entry** and **exit**) and also to code the appropriate transition's effect when necessary.

Topics:

Topic	Detail	See also
Enumerations	<ul style="list-style-type: none"> • StateType - comprises an enumeration for each of the States contained within the State Machine • TransitionType – comprises an enumeration for each transition that has a valid effect associated with it; for example ProcessOrder_Delivered_to_ProcessOrder_Closed • CommandType – comprises an enumeration for each of the behavior types that a State can contain (Do, Entry, Exit) 	State Machine Diagrams ^[815] Java Code Generated From State Machine Diagram ^[1498]
Attributes	<ul style="list-style-type: none"> • currState:StateType - a variable to hold the current State's information • nextState:StateType - a variable to hold the next State's information, set by each State's transitions accordingly • currTransition:TransitionType - a variable to hold the current transition information; this is set if the transition has a valid effect associated with it • transcend:Boolean - a flag used to advise if a transition is involved in transcending between different State Machines (or Submachine states) • xx_history:StateType - a history variable for each State 	State Machine Diagrams ^[815] Java Code Generated From State Machine Diagram ^[1498]

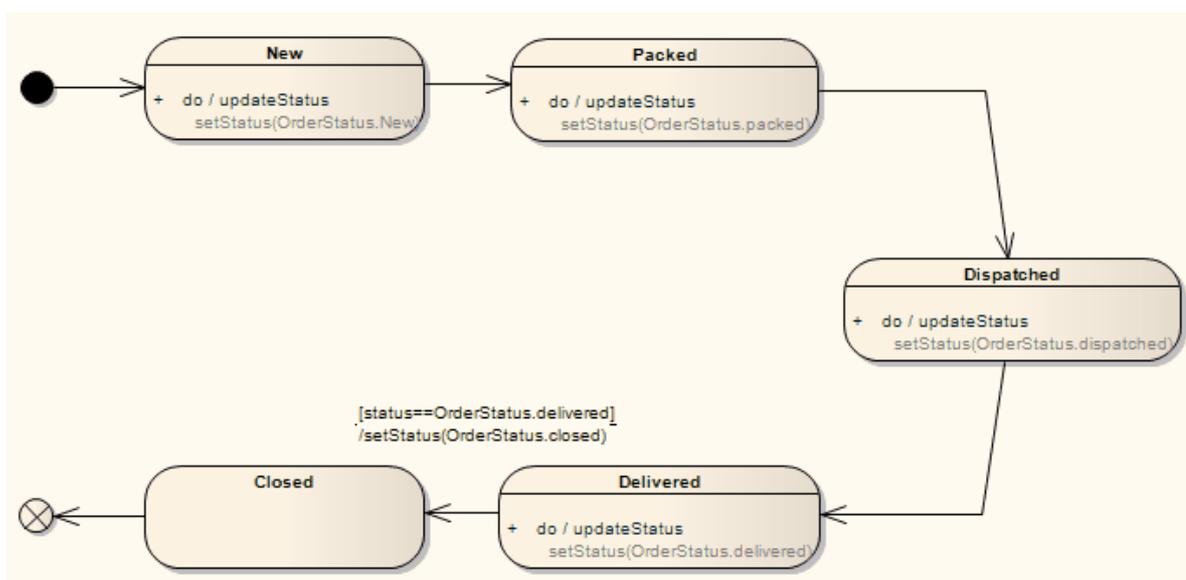
	Machine/Submachine State, to hold information about the last State from which the transition took place	
Operations	<ul style="list-style-type: none"> • StatesProc - a States procedure, containing a map between a State's enumeration and its operation; it de-references the current State's information to invoke the respective State's function • TransitionsProc - a Transitions procedure, containing a map between the Transition's enumeration and its effect; it invokes the respective effect • <<State>> - an operation for each of the States contained within the State Machine; this renders a State's behaviors based on the input CommandType, and also executes its transitions • initializeStateMachine - a function that initializes all the framework-related attributes • runStateMachine - a function that iterates through each State, and executes their behaviors and transitions accordingly 	State Machine Diagrams ^[815] Java Code Generated From State Machine Diagram ^[1498]

Notes:

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class

Learn More:

- [State Machine Diagrams](#) ^[815]
- [Generate From Behavioral Models](#) ^[1498]
- [State Machine Modeling For HDLs](#) ^[1502]

11.6.1.1 Java Code Generated From State Machine Diagram

```

private enum StateType : int
{
    ProcessOrder_Delivered,

```

```

        ProcessOrder_Packed,
        ProcessOrder_Closed,
        ProcessOrder_Dispached,
        ProcessOrder_New,
        ST_NOSTATE
    }
    private enum TransitionType : int
    {
        ProcessOrder_Delivered_to_ProcessOrder_Closed,
        TT_NOTRANSITION
    }
    private enum CommandType
    {
        Do,
        Entry,
        Exit
    }
    private StateType currState;
    private StateType nextState;
    private TransitionType currTransition;
    private boolean transcend;
    private StateType ProcessOrder_history;
    private void processOrder_Delivered(CommandType command)
    {
        switch(command)
        {
            case Do:
            {
                // Do Behaviors..
                setStatus(Delivered);
                // State's Transitions
                if((state==Delivered))
                {
                    nextState = StateType.ProcessOrder_Closed;
                    currTransition = TransitionType.
ProcessOrder_Delivered_to_ProcessOrder_Closed;
                }
                break;
            }
            default:
            {
                break;
            }
        }
    }

    private void processOrder_Packed(CommandType command)
    {
        switch(command)
        {
            case Do:
            {
                // Do Behaviors..
                setStatus(Packed);
                // State's Transitions
                nextState = StateType.ProcessOrder_Dispached;
                break;
            }
            default:
            {
                break;
            }
        }
    }

    private void processOrder_Closed(CommandType command)
    {
        switch(command)
        {
            case Do:
            {

```

```

        // Do Behaviors..
        // State's Transitions
        break;
    }
    default:
    {
        break;
    }
}

private void processOrder_Dispached( CommandType command)
{
    switch( command)
    {
        case Do:
        {
            // Do Behaviors..
            set Status( Dispached);
            // State's Transitions
            next State = StateType. ProcessOrder_Delivered;
            break;
        }
        default:
        {
            break;
        }
    }
}

private void processOrder_New( CommandType command)
{
    switch( command)
    {
        case Do:
        {
            // Do Behaviors..
            set Status( new);
            // State's Transitions
            next State = StateType. ProcessOrder_Packed;
            break;
        }
        default:
        {
            break;
        }
    }
}

private void StatesProc( StateType currState, CommandType command)
{
    switch( currState)
    {
        case ProcessOrder_Delivered:
        {
            processOrder_Delivered( command);
            break;
        }

        case ProcessOrder_Packed:
        {
            processOrder_Packed( command);
            break;
        }

        case ProcessOrder_Closed:
        {
            processOrder_Closed( command);
            break;
        }

        case ProcessOrder_Dispached:

```

```

        {
            processOrder_Di spatched( command);
            break;
        }

    case ProcessOrder_New:
    {
        processOrder_New( command);
        break;
    }

    default:
        break;
    }
}
private void TransitionsProc( TransitionType transition)
{
    switch(transition)
    {
        case ProcessOrder_Delivered_to_ProcessOrder_Closed:
        {
            setStatus( closed);
            break;
        }
        default:
            break;
    }
}
private void initializeStateMachine()
{
    currState = StateType.ProcessOrder_New;
    nextState = StateType.ST_NOSTATE;
    currTransition = TransitionType.TT_NOTRANSITION;
}

private void runStateMachine()
{
    while(true)
    {
        if ( currState == StateType.ST_NOSTATE )
        {
            break ;
        }

        currTransition = TransitionType.TT_NOTRANSITION;
        StatesProc(currState, CommandType.Do);
        // then check if there is any valid transition assigned after the do
behavior
        if ( nextState == StateType.ST_NOSTATE)
        {
            break;
        }

        if ( currTransition != TransitionType.TT_NOTRANSITION )
        {
            TransitionsProc( currTransition );
        }
        if ( currState != nextState)
        {
            StatesProc(currState, CommandType.Exit);
            StatesProc(nextState, CommandType.Entry);
            currState = nextState ;
        }
    }
}
}

```

11.6.2 State Machine Modeling For HDLs

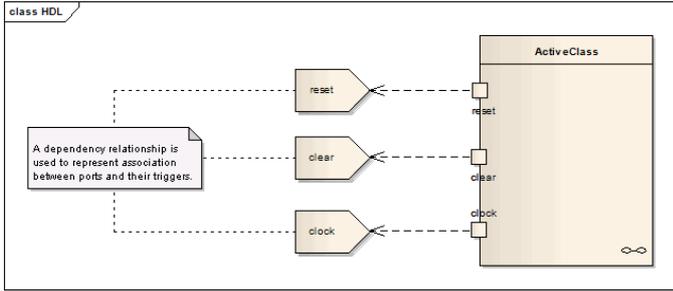
For efficient code generation from State Machine models into Hardware Description Languages (HDL) such as VHDL, Verilog and Systems C, apply the design practices outlined in this topic.

In an HDL State Machine model, the following are expected:

- Designate Driving Triggers
- Establish Port–Trigger Mapping
- Active State Logic

How To:

Topic	Detail	See also																
Designate Driving Triggers	<ul style="list-style-type: none"> • A "change" trigger is deemed as an asynchronous trigger if the following two conditions are satisfied: <ol style="list-style-type: none"> 1. There is a transition from the actual submachine state (which encapsulates the actual logic) triggered by it 2. The target state of that transition has a self transition triggered by the same trigger • Asynchronous triggers should be modeled according to the following pattern: <ol style="list-style-type: none"> 1. The trigger should be of type Change (specification: true / false) 2. The active state (Submachine State) should have a transition trigger by it. 3. The target state of the triggered transition should have a self transition with the same trigger • A trigger of type <i>time</i>, which triggers the transitions to the active state (Submachine State) is deemed as the <i>Clock</i>. The specification of this trigger should be specific to the target language: <table border="1" data-bbox="512 1305 1190 1585"> <thead> <tr> <th rowspan="2">Trigger Type</th> <th rowspan="2">Language</th> <th colspan="2">Specification</th> </tr> <tr> <th>Positive Edge Triggered</th> <th>Negative Edge Triggered</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Time</td> <td>VHDL</td> <td>rising_edge</td> <td>falling_edge</td> </tr> <tr> <td>Verilog</td> <td>posedge</td> <td>negedge</td> </tr> <tr> <td>SystemC</td> <td>positive</td> <td>negative</td> </tr> </tbody> </table> 	Trigger Type	Language	Specification		Positive Edge Triggered	Negative Edge Triggered	Time	VHDL	rising_edge	falling_edge	Verilog	posedge	negedge	SystemC	positive	negative	State Machine Diagrams ^[815] Trigger ^[934]
Trigger Type	Language			Specification														
		Positive Edge Triggered	Negative Edge Triggered															
Time	VHDL	rising_edge	falling_edge															
	Verilog	posedge	negedge															
	SystemC	positive	negative															
Establish Port-Trigger Mapping	<p>After successfully modeling the different operating modes of the component, and the triggers associated with them, you must associate the triggers with the component's ports. A Dependency relationship from the Port to the associated trigger is used to signify their association.</p>																	

		
Active State Logic	The first two aspects, above, put in place the preliminaries required for efficient interpretation of the hardware components. The actual State Machine logic is now modeled within the Active (Submachine) state.	

Notes:

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class
- The current code generation engine supports only one clock trigger for a component

Learn More:

- [SW Code Generation - State Machine Diagrams](#) ^[1497]
- [Transition](#) ^[1013]

11.6.3 Code Generation - Interaction Diagrams

During code generation from interaction(sequence) diagrams in a Class, Enterprise Architect applies its system engineering graph optimizer to transform the Class constructs into programmatic paradigms. Messages and Fragments are identified as one of the several action types based on their functionality, and Enterprise Architect uses the *code generation templates* to render their behavior accordingly.

Topics:

Topic	Detail	See also
Action Call	A Message that invokes an operation	Message (Sequence Diagram) ^[987] Call ^[997] Message Examples ^[992]
Action Create	A Message with <i>Lifecycle = New</i>	Message (Sequence Diagram) ^[987] Message Properties ^[988]
Action Destroy	A Message with Lifecycle = Delete	Message (Sequence Diagram) ^[987] Message Properties ^[988]
Action Loop	A Combined Fragment with Type = Alt	Combined Fragment ^[880] Interaction Operators ^[882]
Action If	A Combined Fragment with Type = loop	Combined Fragment ^[880]

		Interaction Operators ^[882]
Assign To	A call message with a valid target attribute set using the "Assign To" field is rendered in the code as the target attribute of a call action	Call ^[991] Message Properties ^[988]

Notes:

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class
- For an Interaction (Sequence) diagram, the behavioral code generation engine expects the Sequence diagram and all its associated messages and interaction fragments to be encapsulated within an interaction element

Learn More:

- [EASL Code Generation Macros](#) ^[1137]
- [Interaction](#) ^[904]
- [Interaction Occurrence](#) ^[906]
- [Activity](#) ^[873]

11.6.4 Code Generation - Activity Diagrams

Code generation from an Activity Diagrams in a Class requires a validation phase, during which Enterprise Architect uses the system engineering graph optimizer to analyze the diagram and render it into various code-generatable constructs. Enterprise Architect also transforms the constructs into one of the various action types (if appropriate), similar to the Interaction diagram constructs.

Topics:

Topic	Detail	See also
Call Actions (Invocation Actions)	<p>Call Actions are used to invoke operations or behaviors in an Activity diagram. The two main variants of Call Actions supported in behavioral code generation are:</p> <ul style="list-style-type: none"> • CallOperationAction - used to invoke operations, which can be within the same Class or in other Classes within the same package. If referencing operations from other Classes within the same package, you must have a target to which the request is passed • CallBehaviorAction - used to invoke another Activity in an activity flow. The referenced activity is expected to be within the same Class <p>Arguments</p> <p>Call Actions can specify argument values corresponding to the parameters in the associated behavior or behavioral feature. You can add the arguments manually or create them automatically using the the Synchronize button of the Arguments dialog.</p>	Assign Action Pins ^[870] Behavior Calls ^[715] Synchronize Arguments ^[716]
CreateObjectActi	CreateObjectAction is used to denote an object	Assign Action Pins ^[870]

on	<p>creation in the activity flow. You can set the result Pin of the CreateObjectAction as the object to be created, using the Assign Action Pins dialog.</p> <p>The Classifier of the CreateObjectAction signifies the Classifier for which an instance is to be created.</p>	
DestroyObjectAction	<p>DestroyObjectAction is used to denote an object deletion in the activity flow. You can set the target Pin of the DestroyObjectAction as the object to be destroyed, using the Assign Action Pins dialog.</p>	Assign Action Pins ^[870]
Loops	<p>Enterprise Architect's system engineering graph optimizer is also capable of analyzing and identifying loops. An identified loop is internally rendered as an Action Loop, which is translated by the EASL code generation macros to generate the required code</p>	
Conditional Statements	<p>To model a conditional statement, you use Decision/Merge nodes. Alternatively, you can imply Decisions/Merges internally. The graph optimizer expects an associated Merge node for each Decision node, to facilitate efficient tracking of various branches and analysis of the code constructs within them.</p>	

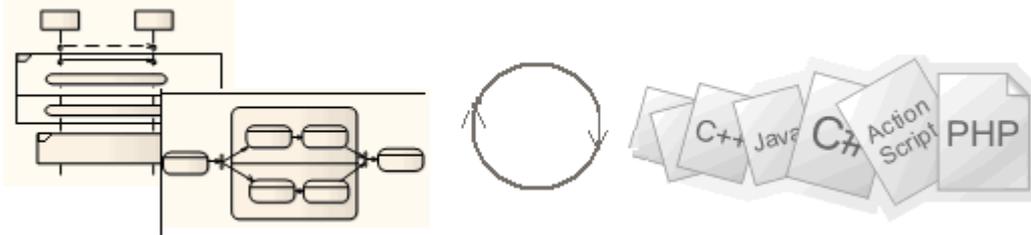
Notes:

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class

Learn More:

- [EASL Code Generation Macros](#) ^[1137]
- [Assign Action Pins](#) ^[870]
- [Interactions and Activities](#) ^[715]
- [Behavior Calls](#) ^[715]
- [Synchronize Arguments](#) ^[716]

11.7 Synchronize Model and Code



In addition to generating and importing code, Enterprise Architect provides the option to synchronize the model and source code, creating a model that represents the latest changes in the source code and vice versa. You can use either the model as the source, or the code as the source.

For example: you generated some source code, but made subsequent changes to the model. When you generate code again, Enterprise Architect *adds* any new attributes or methods to the existing source code, leaving intact what already exists. This means developers can work on the source code and then generate *additional* methods as required from the model, without having their code overwritten or destroyed.

Similarly, you might have made changes to a source code file, but the model has detailed notes and characteristics you do not want to lose. By synchronizing from the source code into the model, you import additional attributes and methods but do not change other model elements.

Using the two synchronization methods above, it is simple to keep source code and model elements up to date and synchronized.

Topics:

Topic	Detail	See also
Synchronize Classes on Forward Generation	<p>When there are features present in the code but not in the model you can use the following buttons during forward synchronization:</p> <ul style="list-style-type: none"> • Delete: when you click on this button the selected code features are removed from the code • Reassign: when you click on this button the code elements are reassigned to elements in the model (this is only possible when an appropriate model element is present that is not already defined in the code) • Ignore: when you click on this button the code elements not present in the model are ignored completely • Reset to Default: when you click on this button the settings for synchronizing during forward generation are set to Ignore, meaning that the elements present in the code but not in the model are ignored completely 	<p>Generate Source Code ^[1490]</p> <p>Generate Source Code and DDL ^[205]</p>

Notes:

- Code synchronization does not change method bodies. Behavioral code generation only works when generating the entire file
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Generate Source Code and DDL permission

to synchronize source code with model elements

- The buttons to influence features generation are only available when the 'On forward synch, prompt to delete code features not in model' checkbox is selected in the Options - Attributes and Operations dialog

Learn More:

- [Generate Source Code](#) [1490]
- [Generate Source Code and DDL](#) [205]
- [Options - Attributes and Operations](#) [1522]
- [Configure User Security](#) [194]
- [Generate From Behavioral Models](#) [1496]

11.8 Importing Source Code



Topics:

Topic	Detail	See also
Usage	<p>Reverse Engineering in Enterprise Architect enables you to import existing source code from a variety of code languages into a UML model. Existing source code structures are mapped into their UML representations, for example a Java Class is mapped into a UML Class element with the variables being defined as attributes, methods are modeled as operations and the interactions between the Java Classes being displayed in the UML model Class diagram with the appropriate connectors.</p> <p>Reverse Engineering enables users to examine legacy code and examine the functionality of code libraries for reuse or to bring the UML model up to date with the code that has been developed as part of a process called <i>synchronization</i>. Examining the code in a UML model enables user to identify the critical modules contained the code, enabling a starting point for understanding of the business and system requirements of the pre-existing system and to enable the developers to gain a better overall understanding of the source code.</p> <p>To begin the process of importing existing code into Enterprise Architect, an existing source of code must be imported into Enterprise Architect, which can be a single directory or a directory structure. Several options are available when performing the reverse engineering process. The Source Code Engineering Options topic contains several options that affect the reverse engineering process. These include:</p> <ul style="list-style-type: none"> • If comments are reverse engineered into notes fields, and how they are formatted if they are • How property methods are recognized • If dependencies should be created for operation return and parameter types <p>It is important to note that when a legacy system is not well designed, simply importing the source into Enterprise Architect does not create an easily understandable UML model. When working with a legacy system that is poorly designed it is useful to break down the code into manageable components by examining the code elements individually. This can be achieved by importing a specific Class of interest into a diagram and then inserting the related elements at one level to determine immediate relationship to other Classes. From this point it is possible to create Use Cases that identify the interaction between the legacy Classes, enabling an overview of the legacy system's operation.</p>	<p>Imported Source Code ^[1510]</p> <p>Import a Directory Structure ^[1513]</p> <p>Source Code Options ^[1517]</p> <p>Insert Related Elements ^[652]</p>
Copyright	Copyright ownership is an important issue to take into account	

Ownership	<p>when undertaking the process of reverse engineering. In some cases, software might have specific limitations that prohibit the process of reverse engineering. It is important that a user address the issue of copyright before beginning the process of reverse engineering code. Situations that typically lend themselves to reverse engineering source code include source code that:</p> <ul style="list-style-type: none"> • You have already developed • Is part of a third-party library that you have obtained permission to use • Is part of a framework that your organization uses • Is being developed on a daily basis by your developers 	
------------------	---	--

Enterprise Architect currently supports reverse engineering in the following programming languages:

Reference:

Topic	Link
Action Script	ActionScript ^[1458]
Ada 2005 (Systems Engineering and Ultimate editions)	Ada 2005 ^[1459]
C	C ^[1461]
C#	C# ^[1463]
C++	C++ ^[1465]
CORBA IDL (MDG Technology)	www.sparxsystems.com/resources/mdg_tech/
Delphi	Delphi ^[1469]
Java	Java ^[1470]
PHP	PHP ^[1472]
Python	Python ^[1473]
SystemC (Systems Engineering and Ultimate editions)	SystemC ^[1474]
Verilog (Systems Engineering and Ultimate editions)	Verilog ^[1477]
VHDL (Systems Engineering and Ultimate editions)	VHDL ^[1478]
Visual Basic	Visual Basic ^[1480]
Visual Basic .NET	Visual Basic .NET ^[1475]

Enterprise Architect is also able to reverse engineer certain types of binary files: Java .jar files and .NET PE files. See **Import Binary Module** for more information.

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Reverse Engineer From DDL And Source**

Code permission to reverse engineer source code and synchronize model elements against code

- Reverse Engineering of other languages is currently available through the use of MDG Technologies from www.sparxsystems.com/resources/mdg_tech/

Learn More:

- [Import Binary Module](#)^[1513]
- [Permission List](#)^[205]
- www.sparxsystems.com/resources/mdg_tech/ (Online Resource)

11.8.1 Import Source Code

To import source code (reverse engineer) follow the steps below

How To:

Step	Action	See also
1	In the Project Browser , select (or add) a diagram into which to import the Classes.	
2	Right-click on the diagram background to open the context menu and either: <ul style="list-style-type: none"> Select the language to import from the Import from source file(s) submenu Click on the Import Language drop-down arrow in the Code Generation toolbar and select the Import Import xxx files menu option, where xxx represents the language to import 	
3	From the file browser that appears, select one or more source code files to import.	Notes on Source Code Import ^[1510]
4	Click on the Open button to start the import process.	

As the import proceeds, Enterprise Architect provides progress information. When all files are imported, Enterprise Architect makes a second pass to resolve associations and inheritance relationships between the imported Classes.

11.8.2 Notes on Source Code Import

Enterprise Architect enables you to **import code** into your project, in the following programming languages:

Topic	Link
Action Script	ActionScript ^[1458]
C	C ^[1461]
C#	C# ^[1463]
C++	C++ ^[1465]
Delphi	Delphi ^[1469]
Java	Java ^[1470]
PHP	PHP ^[1472]

Topic	Link
Python	Python ^[1473]
Visual Basic	Visual Basic ^[1480]
Visual Basic .NET	Visual Basic .NET ^[1475]

Topics:

Topic	Detail	See also
Introduction	<p>Enterprise Architect supports most constructs and keywords for each coding language.</p> <p>If there is a particular feature you require support for that you feel is missing, please contact Sparx Systems.</p> <p>You must select the appropriate type of source file for the language, as the source code to import.</p>	Sparx Systems (Online resource)
ActionScript	Appropriate type of source file: .as .	
C	<p>Appropriate type of source file: .h header files and/or .c files.</p> <p>When you select a header file Enterprise Architect automatically searches for the corresponding .c implementation file to import based on the options for extension and search path specified in the C options.</p> <p>Enterprise Architect does not expand macros that have been used, these must be added into the internal list of Language Macros.</p>	C options ^[1531] Language Macros ^[1525]
C++	<p>Appropriate type of source file: .h header file.</p> <p>Enterprise Architect automatically searches for the .cpp implementation file based on the extension and search path set in the C++ options. When it finds the implementation file it can use it to resolve parameter names and method notes as necessary.</p> <p>When importing C++ source code, Enterprise Architect ignores function pointer declarations. To import them into your model you could create a typedef to define a function pointer type, then declare function pointers using that type. Function pointers declared in this way are imported as attributes of the function pointer type.</p> <p>Enterprise Architect does not expand macros that have been used; these must be added into the internal list of Language Macros.</p>	C++ options ^[1533] Language Macros ^[1525]
C#	Appropriate type of source file: .cs .	
Delphi	Appropriate type of source file: .pas .	
Java	<p>Appropriate type of source file: .java.</p> <p>Enterprise Architect supports the AspectJ language extensions.</p>	

	 <pre> classDiagram class ThingObserving { <<aspect>> -observers: Vector = new Vector() +addObserver(Thing, Thing) : void +removeObserver(Thing, ThingObserver) : void ~updateObserver(Thing, ThingObserver) : void <<advice>> +after(Thing) : void changes(t) <<pointcut>> ~changes(Thing) : void target(t) && call(Void Thing.set*(int)) </pre> <p>Aspects are modeled using Classes with the stereotype aspect. These aspects can then contain attributes and methods as for a normal Class. If an <i>intertype</i> attribute or operation is required, you can add a tag <i>className</i> with the value being the name of the Class it belongs to.</p> <p><i>Pointcuts</i> are defined as operations with the stereotype of <i>pointcut</i>. These can occur in any Java Class, Interface or aspect. The details of the <i>pointcut</i> are included in the behavior field of the method.</p> <p>Advice is defined as an operation with the stereotype advice. The <i>pointcut</i> this advice operates on is in the behavior field and acts as part of the method's unique signature. After advice can also have one of the Tagged Values <i>returning</i> or <i>throwing</i>.</p>	
PHP	Appropriate type of source file: .php, .php4, or .inc. Nested if condition syntax is enabled.	
Python	Appropriate type of source file: .py.	
Visual Basic	Appropriate type of source file: .cls Class file.	
Visual Basic .NET	Appropriate type of source file: .vb Class file.	

Notes:

- When reverse engineering attributes with parameter substitutions (templated attributes):
 - If a Class with proper **template parameter definitions** is found, an Association connector is created and its parameter substitutions are configured
 - An Association connector is also created if a matching entry is defined as a **Collection Class** or in the Additional Collection Classes option (for **C#**, **C++** and **Java**). For an example, see **Example Use of Collection Classes**

Learn More:

- [Import Source Code](#) ^[1510]
- [Parameterized Classes \(Templates\)](#) ^[943]
- [Set Collection Classes](#) ^[1526]
- [C# Options](#) ^[1532]
- [C++ Options](#) ^[1533]
- [Java Options](#) ^[1535]
- [Example Use of Collection Classes](#) ^[1528]

11.8.3 Import a Directory Structure

You can import from all source files in a complete directory structure. This process enables you to import or synchronize multiple files in a directory tree in one pass. Enterprise Architect creates the necessary packages and diagrams during the import process.

To import a directory structure, follow the steps below

How To:

Step	Action	See also
1	In the Project Browser , right-click on the target package for the import.	
2	From the context menu, select the Code Engineering Import Source Directory menu option. The Import Source Directory dialog displays.	
3	Select the options you require. You can configure: <ul style="list-style-type: none"> • The source directory • The source type • The file extensions to look at • Whether to recurse sub directories • Whether to create a diagram for each package • Whether to import additional files as described in the Import Component Types dialog • Whether to exclude private members from libraries being imported from the model • Whether to create a package for every directory, namespace or file; this might be restricted depending on the source type selected • Whether to Synchronize or Overwrite existing Classes when found (if a model Class is found matching the one in code, Synchronize updates the model Class to include the details from the one in code, which preserves information not represented in code such as the location of Classes in diagrams; Overwrite <i>deletes</i> the model Class and generates a new one from code, which deletes and does not replace the additional information) • How to handle Classes not found during the import (Prompt for action enables you to review Classes individually) • What is shown on diagrams created by the import 	Classes not found during Import ^[1513]
4	Click on the OK button to start.	

11.8.4 Import Binary Module

Enterprise Architect enables you to reverse-engineer certain types of binary modules.

Access: Right-click on a package within the Project Browser | Code Engineering | Import Binary Module

Topics:

Topic	Detail	See also

<p>Usage</p>	<p>Currently the permitted types are as follows:</p> <ul style="list-style-type: none"> • Java Archive (.jar) • .Net PE file (.exe, .dll); native Windows DLL and EXE files are not supported, only PE files containing .NET assembly data • Intermediate Language file (.il) <p>Enterprise Architect creates the necessary packages and diagrams during the import process. Selecting the Do not import private members checkbox excludes private members from libraries from being imported into the model.</p> <p>When importing .Net files, you can import via reflection or via disassembly, or let Enterprise Architect decide the best method - this might result in both types being used. The reflection-based importer relies on a .Net program, and requires the .Net runtime environment to be installed. The disassembler-based importer relies on a native Windows program called <i>lldasm.exe</i>, which is a tool provided with the MS .Net SDK. The SDK can be downloaded from the Microsoft website.</p> <p>A choice of import methods is available because some files are not compatible with reflection (such as <i>mscorlib.dll</i>) and can only be opened using the disassembler. However, the reflection-based importer is generally much faster.</p> <p>You can also configure:</p> <ul style="list-style-type: none"> • Whether to Synchronize or Overwrite existing Classes when found (if a model Class is found matching the one in the file, Synchronize updates the model Class to include the details from the one in the file, which preserves information not represented in the file such as the location of Classes in diagrams; Overwrite deletes the model Class and generates a new one from the file, which deletes and does not replace the additional information) • Whether to create a diagram for each package • What is shown on diagrams created by the import 	
---------------------	---	--

11.8.5 MDG Integration and Code Engineering

Topics:

Topic	Detail	See also
<p>Abstract/Usage</p>	<p>MDG Integration for Eclipse and MDG Integration for Visual Studio are standalone products that provide an enhanced code engineering functionality between Enterprise Architect and the development environments.</p> <p>The MDG Integration programs provide a lightweight bridge between Enterprise Architect and the</p>	<p>www.sparxsystems.com/products/mdg/int/eclipse/index.html (Online Resource)</p> <p>www.sparxsystems.com/products/mdg/int/vs/index.html (Online Resource)</p>

	<p>development environment, offering enhanced code generation, reverse engineering and synchronization between code and the UML model. Merging changes can be achieved with minimal effort, and navigation between model and source code is significantly enhanced.</p> <p>A trial version of MDG Integration for Eclipse can be downloaded from www.sparxsystems.com/products/mdg/int/eclipse/index.html and MDG Integration for Visual Studio can be downloaded from www.sparxsystems.com/products/mdg/int/vs/index.html.</p>	
--	--	--

11.8.6 Classes Not Found During Import

Topics:

Topic	Detail	See also
Usage	<p>When reverse synchronizing from your code, there are times when some Classes might be deliberately removed from your source code. Enterprise Architect's import source directory functionality keeps track of the Classes it expects to synchronize with and, on the Import Directory Structure dialog, provides options for how to handle the Classes that weren't found. You can select the appropriate action so that, at the end of the import, Enterprise Architect either ignores the missing Classes, automatically deletes them or prompts you to handle them.</p> <p>If you select the Prompt For Action radio button on the Import Directory Structure dialog, to manually review missing Classes a dialog is displayed allowing you to specify the handling for each class that was missing in the imported code. By default, all Classes are marked for deletion. To keep one or more Classes, select them and click on the Ignore button.</p>	Import a Directory Structure ^[1513]

11.9 Other Settings



You can set the default code options such as the editors for each of the programming languages available for Enterprise Architect and special options for how source code is generated or reverse engineered.

Learn More:

Topic	Details	Link
General Options	Describes the general options that apply to all languages when generating code from Enterprise Architect	General Options <small>[1516]</small>
Local Paths	Describes how to define local paths for each Enterprise Architect user, using the Local Paths settings	Local Paths <small>[1523]</small>
Local Path Dialog	Enables you to set up local paths for a single user on a particular machine	Local Path Dialog <small>[1524]</small>
Language Macros	Describes how to define macros, which is helpful for reverse engineering	Language Macros <small>[1525]</small>
Set Collection Classes	Describes how to define Collection Classes	Set Collection Classes <small>[1526]</small>

11.9.1 Source Code Engineering

The following topics describe general options that apply to all languages when generating code from Enterprise Architect. These options are all available under the Source Code Engineering section of the Options dialog

Access: **Tools | Options | Source Code Engineering**

Topic	Detail	Link
Source Code Options	You can set certain defined options for a particular language while generating the code	Source Code Options <small>[1517]</small>
Options - Code Editors	Describes how to access and configure the source code editor	Options - Code Editors <small>[1519]</small>
Options - Object Lifetimes	Describes configuring various options concerning Object Lifetimes	Options - Object Lifetimes <small>[1521]</small>
Options - Attributes/Operations	Describes configuring Attributes/Operations	Options - Attribute/Operations <small>[1522]</small>
Synchronize Model and Code	Describes synchronizing the model and source code	Synchronize Model and Code <small>[1506]</small>

Topic	Detail	Link
Code Page for Source Editing	Provides an option to define the Unicode character set for code generation	Code Page for Source Editing ^[1523]

11.9.1.1 Source Code Options

When you generate code for a particular language, you can set certain options. These include:

- Create a default constructor
- Create a destructor
- Generate copy constructor
- Select default language
- Generate methods for implemented interfaces
- Set the Unicode options for code generation

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#)

Use to:

- Configure code generation options

Reference:

Field	Usage	See also
Always synchronize with existing file (recommended)	Indicates that imported code synchronizes with an existing file	
Replace (overwrite) existing source file	Indicates that imported code overwrites an existing source file	
Component Types	Allows for the importation of component types, this opens the Import component types dialog	Import Component Types ^[1518]
Default Language for Code Generation:	Indicates the default coding language used in code generation	
Wrap long comment lines at:	Specifies the line length to wrap comments within	
Auto Layout Diagram on Import:	Specifies if and when a diagram is automatically generated on code import	
Output file use both CR & LF	Indicates if carriage return and line feed is a used; ensure that this option is set according to what operating system is currently in use as code might not render correctly	
Prompt when synchronizing (reversing)	Ensures that a prompt displays when synchronization occurs	
Remove hard breaks from comments on import	Indicates that hard breaks are removed from commented sections on importation	

Auto generate role names when creating code	Indicates that role names are generated when creating code	
Do not generate role names when creating code	Indicates that role names are not generated when creating code	
Do not generate members where association direction is "Unspecified"	If association direction is "unspecified", members are not generated	
Create dependencies for operation returns and parameter types	Creates dependencies for operation returns and parameter types	
Comments: Generate	Indicates that comments are generated	
Comments: Reverse	Indicates that reverse comments are generated	
Remove prefixes when generating Get/Set properties	Enables specification of prefixes used in your variable naming conventions, if those prefixes should be removed in the variables' corresponding get/set functions	
Treat as suffixes	Treats the prefixes defined within Remove prefixes when generating Get/Set properties as suffixes	
Capitalized Attribute Name for properties	Capitalizes Attribute names for properties	
Use "Is" for Boolean property Get()	Indicates that the /s keyword is used for the Boolean property Get()	
Code page for source editing	Specifies the character embedding format	

Notes:

- It is worthwhile to configure these settings, as they serve as the defaults for all Classes in the model; you can override these on a per-Class basis using the custom settings (from the Code Generation dialog)

11.9.1.1.1 Import Component Types

The Import Component Types dialog enables you to configure what elements you would like to be created for files of any extension found while importing a source code directory.

Access: **Tools | Options | Source Code Engineering : Component Types**

Use to:

- Configure what elements are to be created for files of any extension found while importing a source code directory

Reference:

Field	Usage	See also
Extension	Indicates the extension name for a component type	
Type	Indicates the type for a component	
Stereotype	Indicates the stereotype name given to a component	
Component List	The defined list of component types	
Save	Saves a component definition and adds it to the component list	
New	Allows a new component type to be defined	
Delete	Deletes a highlighted component from the component list	

Notes:

- You can transport these import component types between models, using the Export Reference Data and Import Reference Data options

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

11.9.1.2 Options - Code Editors

You access the source code editor options via the Code Editors page of the Options dialog. They enable you to configure options for Enterprise Architect's internal editor, as well as the default editor for DDL scripts. You can configure external editors for code languages on each language options page.

Access: **Tools | Options | Source Code Engineering > Code Editors**

Use to:

- Configure options for Enterprise Architect's internal editor
- Configure the default editor for DDL Scripts
- Configure DDL Name Templates

Reference:

Field	Usage	See also
DDL Editor	Specifies what DDL Editor is currently in use; the (...) button opens a dialog that enables you to choose a default DDL editor	
Default Database	Indicates the default database used	
MySQL Storage Engine	Indicates the storage engine used	
DDL Name Templates	Specifies the Foreign Key, Primary Key and Unique Constraint Name templates; the (...) button enables you to specify template names	

Field	Usage	See also
Use inbuilt editor if no external editor set	Specifies the editor for code in a language if no external editor is defined for that language	
Show Line Numbers	Shows line numbers in the editor	
Show Structure Tree	Shows a tree with the results of parsing the open file (requires that the file is parsed successfully)	
Don't parse files larger than	Specifies an upper limit on file size for parsing Use this to prevent performance decrease due to parsing very large files	
Syntax Highlighting Options	Specifies both global and language-specific editor language properties	Editor language properties ^[1520]
Automatically Reverse Engineer on File Save	When this is checked, pressing (Ctrl) + (S) to save in the source code editor automatically reverse engineers the code in the same way as the Save Source and Re-Synchronize Class button does	

11.9.1.2.1 Editor Language Properties

Enterprise Architect enables you to specify syntax highlighting properties for all the programming languages that Enterprise Architect supports at installation.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [Code Editors](#) : [Syntax Highlighting Options](#) (...)

Use to:

- Configure Syntax highlighting for various coding languages

Reference:

Topic	Detail	See also
General Features	<p>The Syntax Highlighting Options button displays the Editor Language properties dialog</p> <p>The panel on the left of the dialog lists the languages for which you can set properties</p> <p>The <i>(Global)</i> item at the top of the list enables you to set properties that apply to all programming languages; however, you can reset a global property to a different value for a specific language, on the page for that language</p> <p>Resetting a global property for one language does not affect that property's value for the other languages</p> <p>Click on the required language in the list, to display the properties for that language:</p> <ul style="list-style-type: none"> • Properties shown in bold indicate that this is the highest level at which this property can be defined (for most language options other than <i>Global</i>, this is effectively the only point at which the property is defined) • Properties shown in normal font are generally the global 	

Topic	Detail	See also
	<p>properties that you can reset just for the current language</p> <p>Scroll through the property categories and individual properties for the language; you can collapse and expand categories as necessary, using the expansion box next to the category name (☐)</p> <p>When you click on a property name, an explanation of that property displays in the panel at the bottom right of the dialog</p>	
Define Properties	<p>To define a property, click on the value field following the property name; depending on the type of property, either the field is enabled for direct editing or a drop-down arrow or Browse button displays (as described for the Tagged Values window) to enable you to select the values to define the property</p> <p>Select or type in the required values</p> <p>The Toolbar icons enable you to:</p> <ul style="list-style-type: none"> • Save your changes to the properties • Reset ALL properties fields to the default settings shipped with Enterprise Architect • Reset the current style field to the default setting (not enabled for non-style fields) 	Tagged Values ^[766]
Assign Keys to Macros	<p>The Macros category enables you to assign (Ctrl+Alt+n) keystroke combinations to coding macros that you have created yourself in the Source Code Viewer</p> <p>When you click on the Browse button in a selected Macro field, the Open Macro dialog displays; this dialog lists the existing macros and, if a key combination has been assigned to a macro, what that key combination is</p> <p>Click on the name of the macro and on the Open button to assign the selected keys to the macro</p>	Source Code Viewer ^[1410]

Notes:

- You cannot currently set properties for any additional languages you include through an MDG Technology
- You can resize this dialog, if required

Learn More:

- [Syntax Highlighting](#)^[1395]
- [MDG Technology](#)^[1070]

11.9.1.3 Options - Object Lifetimes

This set of options enables you to configure various options concerning Object Lifetimes.

Access: **Tools | Options | Source Code Engineering > Object Lifetimes**

Use to:

- Define constructor details when generating code
- Specify whether to create a copy constructor
- Destructor details

Reference:

Field	Usage	See also
Constructor	This set of options specify if a constructor is generated, if a constructor is in-line and the visibility of the default constructor	
Copy Constructor	This set of options specify if a copy constructor is generated, if a copy constructor is in-line and the visibility of the default copy constructor	
Destructor	This set of options specify if a destructor is generated, if a destructor is in-line, if a destructor is defined as <i>virtual</i> and the visibility of the default constructor	

Notes:

- Some options are related to C++ only

11.9.1.4 Options - Attribute/Operations

The Options - Attributes/Operations section enables you to configure Attribute and Operations specifications.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [Attribute/Operations](#)

Use to:

- Configure the default name generated from imported attributes
- Generate methods for implemented interfaces
- Delete model attributes not included in the code during reverse synchronization
- Delete model methods not included in the code during reverse synchronization
- Delete code from features contained in the model during forward synchronization
- Delete model associations and aggregations that correspond to attributes not included in the code during reverse synchronization
- Define whether or not the bodies of methods are included and saved in the Enterprise Architect model when reverse engineering
- Create attributes in quick succession, clearing the dialog when you click on **Save** so that you can enter another attribute name

Reference:

Field	Usage	See also
Attribute Specifications: Default name for associated	Specifies the default name generated from imported attributes	
Attribute Specifications: On reverse synch, delete model attributes not in code	Indicates that on reverse synchronization, model attributes not included within code are removed	

Field	Usage	See also
Attribute Specifications: On reverse synch, delete model associations not in code	Indicates that on reverse synchronization, model associations not included within code are removed	
Operation Specifications: Generate methods for implemented interface	Indicates that methods are generated for an implemented interface	
Operation Specifications: On reverse synch, delete model methods not in code	Indicates that on reverse synchronization, model methods not included within code are removed	
Operation Specifications: Include method bodies in model when reverse engineering	Indicates that on reverse engineering code, method bodies are included within your model	
Options: After Save, re-select edited item	Indicates that after saving, the dialog clears allowing for another attribute name to be specified	
On forward synch, prompt to delete code features not in mode	Indicates that a prompt displays to remove features not in the mode during forward synchronization	

11.9.1.5 Code Page for Source Editing

Enterprise Architect enables you to define the Unicode character set for code generation.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#)

How To:

To set the Unicode character set, follow the steps below:

Step	Action	See also
1	Select the Source Code Engineering option The Source Code Engineering page of the Options dialog displays	
2	In the Code page for source editing field, click on the drop-down arrow and select the appropriate Unicode character set	
3	Click on the Close button	

11.9.2 Local Paths

Access: [Settings](#) | [Local Directories and Paths](#)

Topics:

Topic	Detail	See also
General	Sometimes a team of developers could be working on the same	Local Paths

Topic	Detail	See also
Features	<p>Enterprise Architect model; each developer might store their version of the source code in their local file system, but not always at the same location as their fellow developers</p> <p>To handle this scenario, Enterprise Architect enables you to define local paths for each Enterprise Architect user, using the Local Paths dialog (select the Local Directories and Paths menu option)</p> <p>You can use local paths in generating code and reverse engineering, and in version control, developing XML schemas and generating RTF and HTML reports</p> <p>Local paths take a bit of setting up, but if you want to work collaboratively on source and model concurrently, the effort is well worth while</p> <p>For example:</p> <ul style="list-style-type: none"> • Developer A stores her .java files in a <code>C:\Java\Source</code> directory, while developer B stores his in <code>D:\Source</code> • Both developers want to generate and reverse engineer into the same Enterprise Architect model located on a shared (or replicated) network drive <p>Developer A might define a local path of:</p> <pre>JAVA_SOURCE = " C: \ J a v a \ S o u r c e "</pre> <p>All Classes generated and stored in the Enterprise Architect project are stored as:</p> <pre>%J A V A _ S O U R C E % \ < x x x . j a v a ></pre> <p>Developer B defines a local path as:</p> <pre>J A V A _ S O U R C E = " D : \ S o u r c e "</pre> <p>Now, Enterprise Architect stores all java files in these directories as:</p> <pre>%J A V A _ S O U R C E % \ < f i l e n a m e ></pre> <p>On each developer's machine, the filename is expanded to the correct local version</p>	Dialog ¹⁵²⁴

11.9.3 Local Paths Dialog

The Local Paths dialog enables you to set up local paths for a single user on a particular machine. For a description of what Local Paths are used for, see the *Local Paths* topic.

Access: [Settings | Local Directories and Paths](#)

Topics:

Topic	Detail	See also
General Usage	<p>The Local Paths dialog enables you to define:</p> <ul style="list-style-type: none"> • Path - the local directory in the file system (for example, <code>d:\java\source</code>) • ID - the shared ID that is substituted for the Local Path (for example, <code>JAVA_SRC</code>) • Type - the language type (for example, <code>Java</code>) 	

Topic	Detail	See also
	<p>And also:</p> <ul style="list-style-type: none"> • Apply Path - select a path and click on this button to update any existing paths in the model (with full path names) to the shared relative path name (for example: d: \ j a v a \ s o u r c e \ m a i n . j a v a might become % J A V A _ S R C % \ m a i n . j a v a) • Expand Path - the opposite of Apply Path; this enables you to remove a relative path and substitute the full path name <p>Using the two options you can update and change existing paths</p>	

Notes:

- You can also set up a hyperlink on a diagram to access the Local Paths dialog, to switch, update or expand your current Local Path

Learn More:

- [Local Paths](#)^[1523]
- [Hyperlink](#)^[1292]

11.9.4 Language Macros

When reverse engineering a language such as C++, you might find preprocessor directives scattered throughout the code. This can make code management easier, but can hamper parsing of the underlying C++ language.

To help remedy this, you can include any number of *macro* definitions, which are ignored during the parsing phase of the reverse engineering. It is still preferable, if you have the facility, to preprocess the code using the appropriate compiler first; this way, complex macro definitions and defines are expanded out and can be readily parsed. If you don't have this facility, then this option provides a convenient substitute.

Access: [Settings | Preprocessor Macros](#)

How To:

To define a macro, follow the steps below:

Step	Action	See also
1	Select the Preprocessor Macros menu option The Language Macros dialog displays	
2	Click on the Add New button	
3	Enter details for your macro	
4	Click on the OK button	

Topics:

Topic	Detail	See also
Macros Embedded Within Declarations	<p>Macros are sometimes used within the declaration of Classes and operations, as in the following examples:</p> <pre>class __declspec Foo { int __declspec Bar(int p); };</pre> <p>If <i>declspec</i> is defined as a C++ macro, as outlined above, the imported Class and operation contain a Tagged Value called <i>DeclMacro1</i> with value <i>__declspec</i> (subsequent macros would be defined as <i>DeclMacro2</i>, <i>DeclMacro3</i> and so on)</p> <p>During forward engineering, these Tagged Values are used to regenerate the macros in code</p>	
Define Complex Macros	<p>It is sometimes useful to define rules for complex macros that can span multiple lines;Enterprise Architect ignores the entire code section defined by the rule</p> <p>Such macros can be defined in Enterprise Architect as in the following two examples; both types can be combined in one definition</p> <p>Block Macros</p> <pre>BEGIN_INTERFACE_PART ^ END_INTERFACE_PART</pre> <p>where the ^ symbol represents the body of the macro - this enables skipping from one macro to another; the spaces surrounding the ^ symbol are required</p> <p>Function Macros</p> <pre>RTTI_EMULATION()</pre> <p>where Enterprise Architect skips over the token including everything inside the parentheses</p>	

Notes:

- You can transport these language macro (or preprocessor macro) definitions between models, using the **Export Reference Data** and **Import Reference Data** options; the macros are exported as a Macro List

Learn More:

- [Export Reference Data](#)^[237]
- [Import Reference Data](#)^[239]

11.9.5 Set Collection Classes

Enterprise Architect enables you to define *Collection Classes* for generating code from Association connectors where the target role has a multiplicity setting greater than 1.

Topic	Detail	See Also
Defining Collection Classes	On the Source Code Engineering section of the Options dialog (select the Tools Options Source Code Engineering option), on each language page	Source Code Options ^[151]

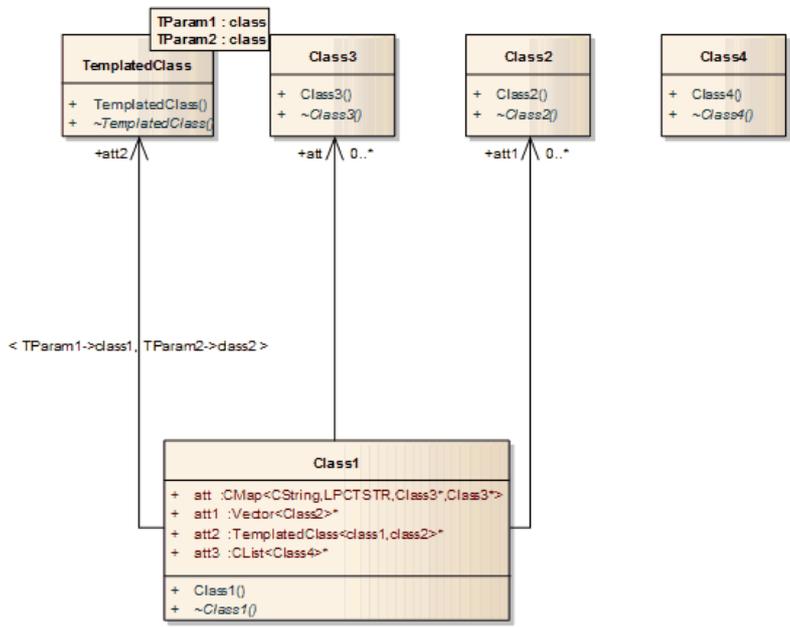
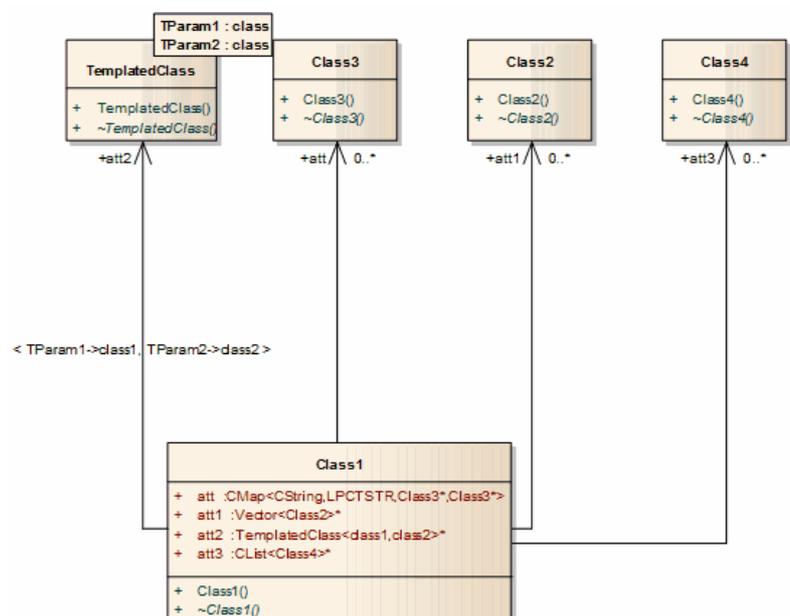
Topic	Detail	See Also
	<p>click on the Collection Classes button</p> <p>The Collection Classes for Association Roles dialog displays</p> <p>On this dialog, you can define:</p> <ul style="list-style-type: none"> • The default Collection Class for 1..* roles • The ordered Collection Class to use for 1..* roles • The qualified Collection Class to use for 1..* roles 	
<p>Defining Collection Classes for a specific Class</p>	<p>Class-specific Collection classes can be defined by clicking the Collection Classes button in the Class Properties dialog of the element</p>	<p>Element Details <small>[663]</small></p>
<p>Code Generation Precedence</p>	<p>When Enterprise Architect generates code for a connector that has a multiplicity role >1:</p> <ol style="list-style-type: none"> 1. If the Qualifier is set use the qualified collection: <ul style="list-style-type: none"> • for the Class if set • else use the code language qualified collection 2. If the Order option is set use the ordered collection: <ul style="list-style-type: none"> • for the Class if set • else use the code language ordered collection 3. Else use the default collection: <ul style="list-style-type: none"> • for the Class if set • else use the code language default collection 	
<p>Using Markers</p>	<p>You can include the marker #TYPE# in the collection name; Enterprise Architect replaces this with the name of the Class being collected at source generation time (for example, <i>Vector<#TYPE#></i> would become <i>Vector<foo></i>)</p> <p>Conversely, when reverse engineering, an Association connector is also created if a matching entry (e.g. <i>foo</i> if <i>foo</i> is found in the model) is defined as a Collection Class</p>	
<p>Additional Collection Classes</p>	<p>Additional Collection Classes can be defined in the language options page (C#, C++, Java)</p>	<p>Example Use of Collection Classes <small>[1528]</small></p> <p>C# Options <small>[1532]</small></p> <p>C++ Options <small>[1533]</small></p> <p>Java Options <small>[1535]</small></p>
<p>Member Type</p>	<p>On both the Source Role and Target Role tabs of the Association Property dialog (accessible from the right-click context menu of any Association) there is a Member Type field</p> <p>If you set this, the value you enter overrides all the above options</p>	<p>Source Role <small>[758]</small></p> <p>Target Role <small>[760]</small></p>

Learn More:

- [Generate Source Code](#) ^[1490]
- [Importing Source Code](#) ^[1508]
- [Source Code Options](#) ^[1517]

11.9.5.1 Example Use of Collection ClassesTopics:

Topic	Detail	See also
Example Code	<p>Consider the following source code:</p> <pre> class Class1 { public: Class1(); virtual ~Class1(); CMap<CString, LPCTSTR, Class3*, Class3*> att; Vector<Class2> *att1; TemplatedClass<class1, class2> *att2; CList<Class4> *att3; }; class Class2 { public: Class2(); virtual ~Class2(); }; class Class3 { public: Class3(); virtual ~Class3(); }; class Class4 { public: Class4(); virtual ~Class4(); }; template<class TParam1, class TParam2> class TemplatedClass { public: TemplatedClass() { } virtual ~TemplatedClass() { } }; </pre>	
Generated Diagram	<p>If this code above is imported into Enterprise Architect with default import options, the following diagram is generated:</p>	<p>C# Options ^[1532]</p> <p>Java</p>

Topic	Detail	See also
	 <p>If, however, you enter the value CList<#Type#> in the Additional Collection Classes field in the language options page (C#, Java, C++), an Association connector is also created to Class 4:</p> 	<p>Options <small>[1535]</small></p> <p>C++ Options <small>[1533]</small></p>

11.9.6 Language Options

You can set up various options for how Enterprise Architect handles a particular language when generating code. These options are accessible on the Options dialog (select the **Tools | Options** menu option).

Under the **Source Code Engineering** option, select the required language. The following topics outline the options available for each language.

Topic	Link
Action Script	ActionScript ^[1530]
Ada 2005 (in the Systems Engineering and Ultimate editions of Enterprise Architect)	Ada 2005 ^[1531]
ANSI C	ANSI C ^[1531]
C#	C# ^[1532]
C++	C++ ^[1533]
Delphi	Delphi ^[1533]
Delphi Properties	Delphi Properties ^[1534]
Java	Java ^[1535]
PHP	PHP ^[1536]
Python	Python ^[1536]
SystemC	SystemC ^[1537]
Verilog (Systems Engineering and Ultimate editions)	Verilog ^[1538]
VHDL (Systems Engineering and Ultimate editions)	VHDL ^[1538]
Visual Basic	Visual Basic ^[1539]
Visual Basic .NET	Visual Basic .NET ^[1537]
MDG Technology Languages	MDG Technology Languages ^[1540]
Reset Options	Reset Options ^[1540]

11.9.6.1 ActionScript Options

Configure options for ActionScript code generation using the ActionScript Specifications page of the Options dialog.

Access: **Tools | Options | Source Code Engineering > ActionScript**

Use to:

- Specify default ActionScript version to generate (AS2.0 or AS3.0)
- Specify default file extensions (header and source)
- Specify default source directory
- Specify the editor for ActionScript code

Reference:

Field	Usage	See also
Disable Language	Indicates that the ActionScript is to be disabled	

Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes [1526]

11.9.6.2 Ada 2005 Options

Configure options for Ada 2005 code generation using the Ada page of the Options dialog.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [Ada](#)

Use to:

- Inform the reverse engineering process whether the name of the Tagged Record is the same as the package name
- Advise the engine of the alternate Tagged Record name to locate
- Specify whether the engine should create a reference type for the Tagged Record (if one is not defined)
- Supply the name of the reference type to be created (default is Ref)
- To specify the reference parameter of a Reference / Access type
- Tell the engine to ignore the name of the reference parameter
- Indicate the name of the reference parameter to locate

Reference:

Field	Usage	See also
Disable Language	Indicates that Ada is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes [1526]

Notes:

- Ada 2005 support is available in the System Engineering and Ultimate editions of Enterprise Architect

11.9.6.3 C Options

Configure options for C code generation using the C Specifications page of the Options dialog.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [C](#)

Use to:

- Support Object Oriented coding
- Indicate default file extensions (header and source)
- Indicate default source directory
- Indicate an editor for C code
- Specify a path that Enterprise Architect uses to search for the implementation file; the first path in the list is the default path when generating

Reference:

Field	Usage	See also
Disable Language	Indicates that C is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1526]</small>

11.9.6.4 C# Options

Configure options for C# code generation using the C# Specifications page of the Options dialog.

Access: **Tools | Options | Source Code Engineering > C#**

Use to:

- Indicate the default File extension
- Indicate additional Collection Classes - to define custom collection classes, which can be simple substitutions (such as *CArray<#TYPE#>*) or a mix of other strings and substitutions (such as *Cmap<CString,LPCWSTR,#TYPE#,#TYPE#*>*); the following collection classes are defined by default:
 - *List<#TYPE#>;Stack<#TYPE#>;Queue<#TYPE#>;*
- Specify a 'Get' prefix
- Specify a 'Set' prefix
- Specify a directory for opening and saving C# source code

Reference:

Field	Usage	See also
Disable Language	Indicates that C# is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	

Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1526]</small>

11.9.6.5 C++ Options

Configure options for C++ code generation using the C++ Specifications page of the Options dialog.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [C++](#)

Use to:

- Indicate the version of C++ to generate; this controls the set of templates used and how properties are created
- Specify the default reference type used when a type is specified by reference
- Specify the default file extensions
- Specify default Get/Set prefixes
- Define additional Collection Classes - to define custom collection classes, which can be simple substitutions (such as `CArray<#TYPE#>`) or a mix of other strings and substitutions (such as `Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>`); the following collection classes are defined by default:
 - `CArray<#TYPE#>;CMap<CString,LPCTSTR,#TYPE#*,#TYPE#*>;`
- Specify a default source directory
- Specify the path that Enterprise Architect uses to search for the implementation file; the first path in the list is the default path when generating new implementation files and parsing existing files - if you add further directories, Enterprise Architect also searches these when parsing existing files

Reference:

Field	Usage	See also
Disable Language	Indicates that C++ is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1526]</small>

11.9.6.6 Delphi Options

Configure options for Delphi code generation using the Delphi Specifications page of the Options dialog

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [Delphi](#)

Use to:

- Indicate a default file extension
- Indicate a default source directory
- Set a default directory for opening and saving Delphi source code

Reference:

Field	Usage	See also
Disable Language	Indicates that Delphi is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes 

Learn More:

- [Delphi Properties](#) 

11.9.6.6.1 Delphi Properties

Enterprise Architect has comprehensive support for Delphi properties. These are implemented as Tagged Values, with a specialized property editor to help create and modify Class properties. By using the **Feature Visibility** element context menu option, you can display the 'tags' compartment that contains the properties. Imported Delphi Classes with properties have this feature automatically made visible for your convenience.

Use to:

- Configure additional properties for Delphi based elements

Reference:

Topic	Description	See also
To manually activate the property editor	<ul style="list-style-type: none"> • Ensure the Class you have selected has the code generation language set to Delphi • Right-click on the Class and select the Delphi Properties context menu option to open the editor <p>The Delphi Properties editor enables you to build properties in a simple and straightforward manner; from here you can:</p> <ul style="list-style-type: none"> • Change the name and scope (only Public and Published are currently supported) • Change the property type (the drop-down list includes all defined Classes in the project) • Set the Read and Write information (the drop-down lists have all the attributes and operations from the current 	

	Class; you can also enter free text) <ul style="list-style-type: none"> • Set Stored to True or False • Set the Implements information • Set the default value, if one exists 	
--	---	--

Notes:

- When you use the Create Property dialog from the Attribute screen, Enterprise Architect generates a pair of Get and Set functions, together with the required property definition as Tagged Values; you can manually edit these Tagged Values if required
- Public properties are displayed with a '+' symbol prefix and published with a '^'
- When creating a property in the Create Property Implementation dialog (accessed through the Attributes dialog), you can set the scope to Published if the property type is Delphi
- Only **Public** and **Published** are supported
- If you change the name of a property and forward engineer, a new property is added, but the old one must be manually deleted from the source file

11.9.6.7 Java Options

Configure options for Java code generation using the Java Specifications page of the Options dialog.

Access: **Tools | Options | Source Code Engineering > Java**

Use to:

- Set a default directory for opening and saving Java source code
- Indicate a default file extension
- Specify a default 'Get' prefix
- Specify a default 'Set' prefix
- Define additional Collection Classes - to define custom collection classes, which can be simple substitutions (such as *CArray<#TYPE#>*) or a mix of other strings and substitutions (such as *Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>*); the following collection classes are defined by default:
 - *HashSet<#TYPE#>;Map<String,#TYPE#>;*

Reference:

Field	Usage	See also
Disable Language	Indicates that Java is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1528]</small>

11.9.6.8 PHP Options

Configure options for PHP code generation using the PHP Specifications page of the Options dialog.

Access: [Tools | Options | Source Code Engineering > PHP](#)

Use to:

- Specify the extension to be used when creating files for PHP source
- Define a semi-colon separated list of extensions to look at when doing a directory code import for PHP
- Set a default directory for opening and saving PHP source code
- Specify the version of PHP to generate

Reference:

Field	Usage	See also
Disable Language	Indicates that PHP is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	

Learn More:

- [Import a Directory Structure](#) ^[1513]

11.9.6.9 Python Options

Configure options for Python code generation using the Python Specifications page of the Options dialog.

Access: [Tools | Options | Source Code Engineering > Python](#)

Use to:

- Indicate the default file extension(s)
- Specify the default source directory to be used
- Specify the editor used to write code

Reference:

Field	Usage	See also
Disable Language	Indicates that Python is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	

Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
-------------------------------------	--	--

11.9.6.10 SystemC Options

Configure options for SystemC code generation using the SystemC page of the Options dialog.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [SystemC](#)

Use to:

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

Reference:

Field	Usage	See also
Disable Language	Indicates that SystemC is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1526]</small>

11.9.6.11 VB.NET Options

Configure options for VB.NET code generation using the VB.NET Specifications page of the Options dialog.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [VB.NET](#)

Use to:

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

Reference:

Field	Usage	See also
Disable Language	Indicates that VB.NET is to be disabled	
Options for the	Specifies the options used in the current model	

current model	These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1526]</small>

11.9.6.12 Verilog Options

Configure options for Verilog code generation using the Verilog page of the Options dialog.

Access: [Tools | Options | Source Code Engineering > Verilog](#)

Use to:

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

Reference:

Field	Usage	See also
Disable Language	Indicates that Verilog is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1526]</small>

11.9.6.13 VHDL Options

Configure options for VHDL code generation using the VHDL page of the Options dialog.

Access: [Tools | Options | Source Code Engineering > VHDL](#)

Use to:

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

Reference:

Field	Usage	See also
Disable Language	Indicates that VHDL is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current user; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1526]</small>

11.9.6.14 Visual Basic Options

Configure options for Visual Basic code generation using the VB Specifications page of the Options dialog.

Access: [Tools](#) | [Options](#) | [Source Code Engineering](#) > [Visual Basic](#)

Use to:

- Indicate the default file extension when reading/writing
- Specify the default Visual Basic version
- Indicate the MTS transaction mode for MTS objects
- Specify if a Class uses Multi use (**true** or **false**)
- Specify if a Class uses the **Persistable** property
- Indicate data binding behaviours
- Set the global namespace
- Set the *Exposed* attribute
- Indicate if the *Creatable* attribute is **true** or **false**

Reference:

Field	Usage	See also
Disable Language	Indicates that Visual Basic is to be disabled	
Options for the current model	Specifies the options used in the current model These options affect all users of the current model; however, they do not apply to other models	
Options for the current user	Specifies the options used for the current use; these options apply to all models that are accessed by the user	
Collection Classes	Opens the Collection Classes for Association Roles dialog, which allows the specification of collection class definitions	Collection Classes <small>[1526]</small>

11.9.6.15 MDG Technology Language Options

If you have loaded an MDG Technology that specifies a code module into your *Sparx Systems > EA > MDG Technologies* folder, the language is included in the Source Code Engineering list on the Options dialog. The language is only listed on the Options dialog if an MDG Technology file actually uses it in your model.

Access: [Tools | Options | Source Code Engineering > MDG](#)

Reference:

Field	Usage	See also
Default Extension	Default extension for generated source files; shown if the option is in the technology This is saved per project	
Import File Extensions	Default folder to import source files from; shown if the technology supports namespaces This is saved once for all projects	
Generate Namespaces	Indicates if namespaces are generated or not	
Default Source Directory	The default directory to save generated source files This is always shown	
Editor	Indicates the editor that is used to edit source files	
Att Type	Indicates the default attribute type	

Notes:

- These options are set in the technology inside the `<CodeOptions>` tag of a code module, as follows:

```
<CodeOption name="DefaultExtension" >.rb</CodeOption>
```

Learn More:

- [MDG Technology](#)^[1064]
- [Adding Code Modules in MDG Technologies](#)^[1070]

11.9.6.16 Reset Options

Enterprise Architect stores some of the options for a Class when it is first created. Some are global; for example, `$LinkClass` is stored when you first create the Class, so it won't automatically pick up the global change in the Options dialog in existing Classes. You must modify the options for the existing Class.

How To:

To modify options for a single Class, follow the steps below:

Step	Action	See also
1	Right-click on the Class to change, and select the Generate Code context menu option The Generate Code dialog displays	
2	Click on the Advanced button The Object Options dialog displays	
3	Click on the Attributes/Operations button	
4	Change the options, and click on the Close button to apply the changes	

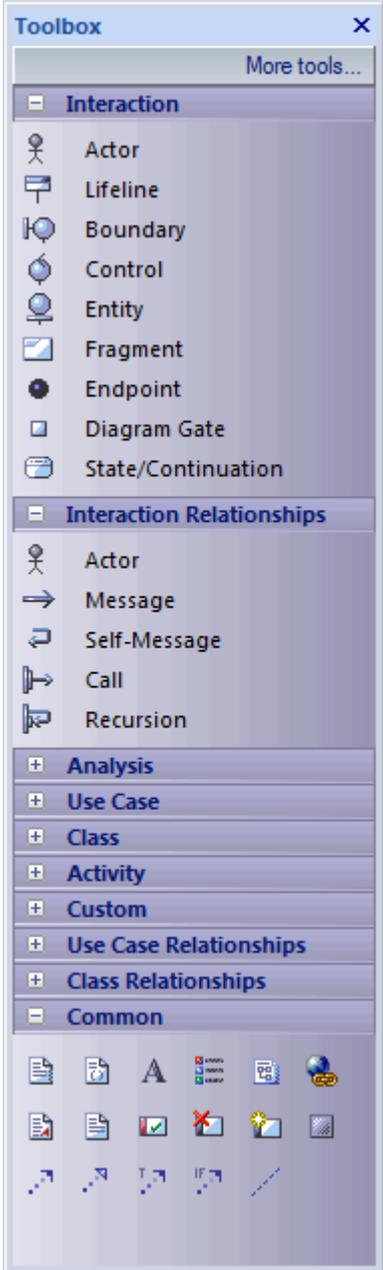
To modify options for all Classes within a package, follow the steps below:

Step	Action	See also
1	Right-click on the package in the Project Browser The context menu displays	
2	Select the Code Engineering Reset Options for this Package menu option The Manage Code Generation dialog displays	
3	Reset the required defaults for each existing Class	
4	Click on the OK button to apply the changes	

11.10 ICONIX

Topics:

Topic	Detail	See also
Abstract	<p>The following text is derived from the ICONIX entry in the online Wikipedia:</p> <p><i>The ICONIX Process is a minimalist, streamlined approach to Use Case driven UML modeling that uses a core subset of UML diagrams and techniques to provide thorough coverage of object-oriented analysis and design. Its main activity is robustness analysis, a method for bridging the gap between analysis and design. Robustness analysis reduces the ambiguity in use case descriptions, by ensuring that they are written in the context of an accompanying domain model. This process makes the use cases much easier to design, test and estimate.</i></p> <p>The ICONIX Process was developed by Doug Rosenberg; for more information on ICONIX, refer to ICONIX Software Engineering Inc.</p>	ICONIX (Online Resource)
ICONIX in Enterprise Architect	<p>Enterprise Architect enables you to develop models under ICONIX quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer</p> <p>The ICONIX facilities are provided in the form of:</p> <ul style="list-style-type: none"> • A set of ICONIX pages in the Toolbox • ICONIX element and relationship entries in the Toolbox Shortcut Menu and Quick Linker <p>To further help you develop and manage a project under ICONIX, Enterprise Architect also provides a white paper on the ICONIX Roadmap</p> <p>In addition, Enterprise Architect has an alternative visual layout specific to ICONIX</p>	Toolbox Shortcut Menu ^[55] Quick Linker ^[62] ICONIX Roadmap (Online Resource)
ICONIX Toolbox Pages	<p>Within the Toolbox, Enterprise Architect provides ICONIX versions of the pages for UML Analysis, Use Case, Class, Interaction (Sequence), Activity and Custom diagrams (which often form the basis for <i>Robustness diagrams</i>)</p> <p>Compared to the standard Toolbox pages, these have slightly different element and relationship sets; you can access them by either:</p> <ul style="list-style-type: none"> • Selecting the More tools ICONIX <Diagram Type> menu option for a specific Toolbox page • Selecting the ICONIX option in the drop-down field of the Default Tools toolbar, which adds all six pages to the Toolbox; the first page and the Common page are expanded, and the others are closed up 	Analysis Diagram ^[118] Use Case Diagram ^[81] Class Diagram ^[79] Sequence Diagram ^[84] Activity Diagram ^[81] Custom Diagram ^[118]

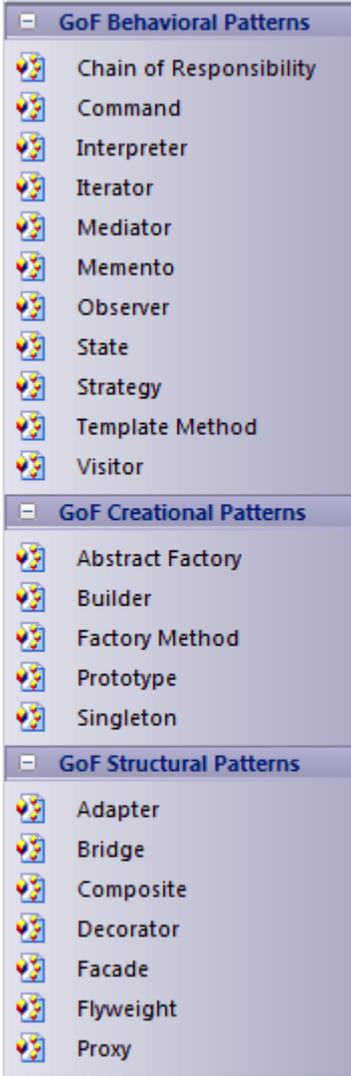
Topic	Detail	See also
		
ICONIX Layout	<p>The ICONIX layout re-organizes the Enterprise Architect work area, opening the:</p> <ul style="list-style-type: none"> • Toolbox on the right hand side of the screen (follow the instructions above to display the ICONIX pages) • The Learning Center window auto-hidden in the top right of the screen • Project Browser window in the top left of the screen, and • Notes, Properties and Tagged Values windows nested on the bottom left of the screen <p>To apply this layout, select the View Workspace Layouts menu option and select the Basic Layout (Alternate) option</p>	Workspace Layout [115]
Disable ICONIX	<p>If you prefer not to use ICONIX in Enterprise Architect, you can</p>	MDG Technologies

Topic	Detail	See also
	<p>disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies)</p> <p>This does not affect the ICONIX layout, which you can switch back to your own layout or the Enterprise Architect default layout using the View Workspace Layouts menu option</p>	 1033

11.11 GoF Patterns

Topics:

Topic	Detail	See also
Abstract	<p>The following text is derived from the Design Patterns entry in the online Wikipedia.</p> <p><i>Gang of Four (GoF) Patterns are 23 classic software design patterns providing recurring solutions to common problems in software design. They were developed by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, often referred to as the Gang of Four. The patterns are defined in the book Design Patterns: Elements of Reusable Object-Oriented Software (Gamma et al., ISBN 0-201-63361-2).</i></p> <p>For further information on the concepts of GoF Patterns, refer to the Wikipedia item and its linked sources.</p>	Design Patterns (Online Resource)
GoF Patterns in Enterprise Architect	<p>Enterprise Architect enables you to develop diagrams from GoF patterns quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The GoF Patterns are provided in the form of:</p> <ul style="list-style-type: none"> • GoF Behavioral Patterns, GoF Creational Patterns and GoF Structural Patterns pages in the Toolbox • Gang of Four pattern entries in the Toolbox Shortcut Menu. <p><u>GoF Toolbox Pages</u></p> <p>You can access the GoF Pattern pages of the Toolbox through the More tools GoF Patterns menu option. The following icons are available:</p>	Toolbox Shortcut Menu ^[55] Use A Pattern ^[102]

Topic	Detail	See also
	 <p>When you drag one of the pattern elements onto a new diagram, the Add Pattern GoF <pattern group><pattern type> dialog displays. If necessary, modify the action and/or default for the component elements, then click on the OK button to create a diagram based on the pattern.</p> <p>The GoF patterns are drawn from the Resources window. If you delete a pattern in the Resources window the equivalent Toolbox item cannot work. Therefore, if you cannot drop a pattern element from the Toolbox, check that it is still available in the Resources window.</p>	
Disable GoF Patterns	If you prefer not to use the GoF Patterns technology in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).	MDG Technologies <small>[1033]</small>

Part

XII

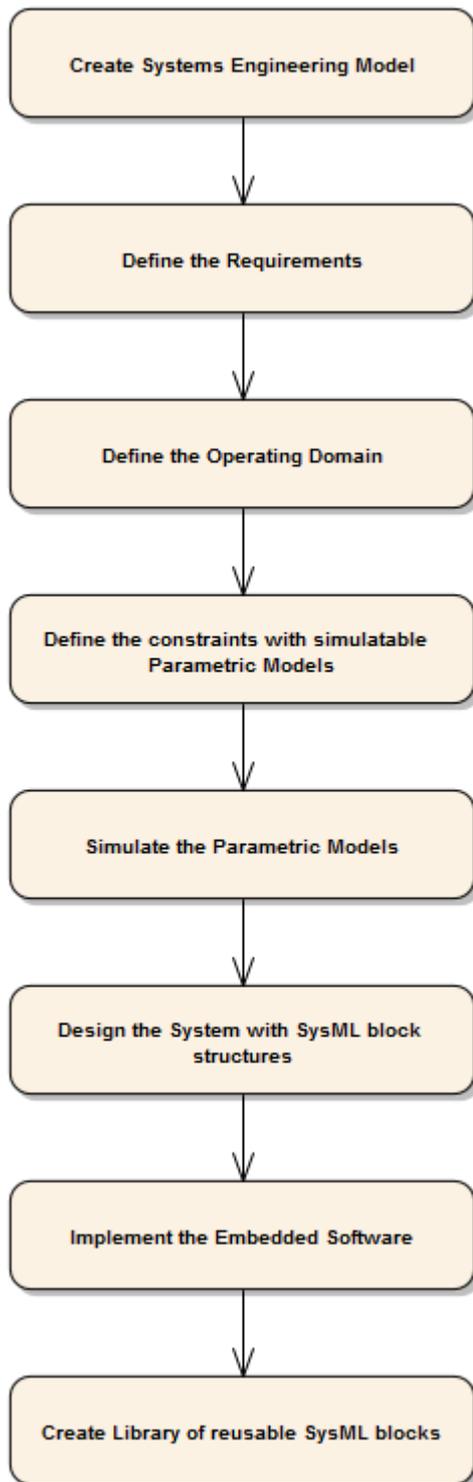
12 Systems Engineering

Topics:

Topic	Detail	See also
Usage	<p>To model Systems using SysML in Enterprise Architect, you work through the following steps:</p> <ul style="list-style-type: none"> • Create a Systems Engineering model to develop your system • Create a Requirements model to define the systems requirements and expectations • Create an Operational Domain model, which describes the environment that the system operates within, and the entities it interacts with • Create Constraint models to describe the systems operating characteristics using parametric models • Simulate the parametric models to verify their correctness and obtain the desired characteristic • Design the system's composition using SysML Blocks and Parts • Implement the embedded software using UML Classes and behavioral models • Create a Library of reusable SysML blocks, representing subsystems that can be reused on other projects, and other common Type definitions 	<p>SysML ^[1552]</p> <p>Create a Systems Engineering model ^[1549]</p> <p>Create a Requirements model ^[1572]</p> <p>Create an Operational Domain model ^[1573]</p> <p>SysML Parametric Models ^[1568]</p> <p>Simulating a SysML Model ^[1570]</p> <p>Compose System Design ^[1575]</p> <p>Generate from Behaviour Models ^[1496]</p> <p>Create Re-usable Subsystems ^[1577]</p>

Example:

These steps are represented graphically in the following flow:



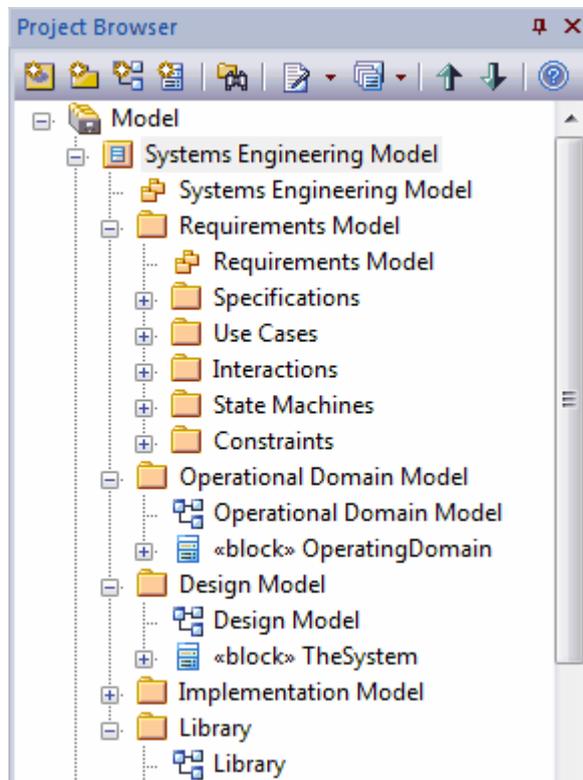
To create a Systems Engineering Model from a template provided with Enterprise Architect, follow the steps below

How To:

Step	Action	See Also
1.	In the Project Browser , either: <ul style="list-style-type: none"> Click on the New Model From Pattern icon in the toolbar Right-click on a model root node and select the Add a New Model using Wizard context menu option, or Right-click on a package and select the Add Add a New Model using Wizard context menu option. The Select Model(s) dialog displays.	
2.	In the Select From field, click on the drop-down arrow and select Systems Engineering Model . Alternatively, if it is listed in the Technology panel, select the Systems Engineering Model item.	
3.	In the Name panel, select the checkbox next to the Systems Engineering Model icon.	
4.	Click on the OK button.	

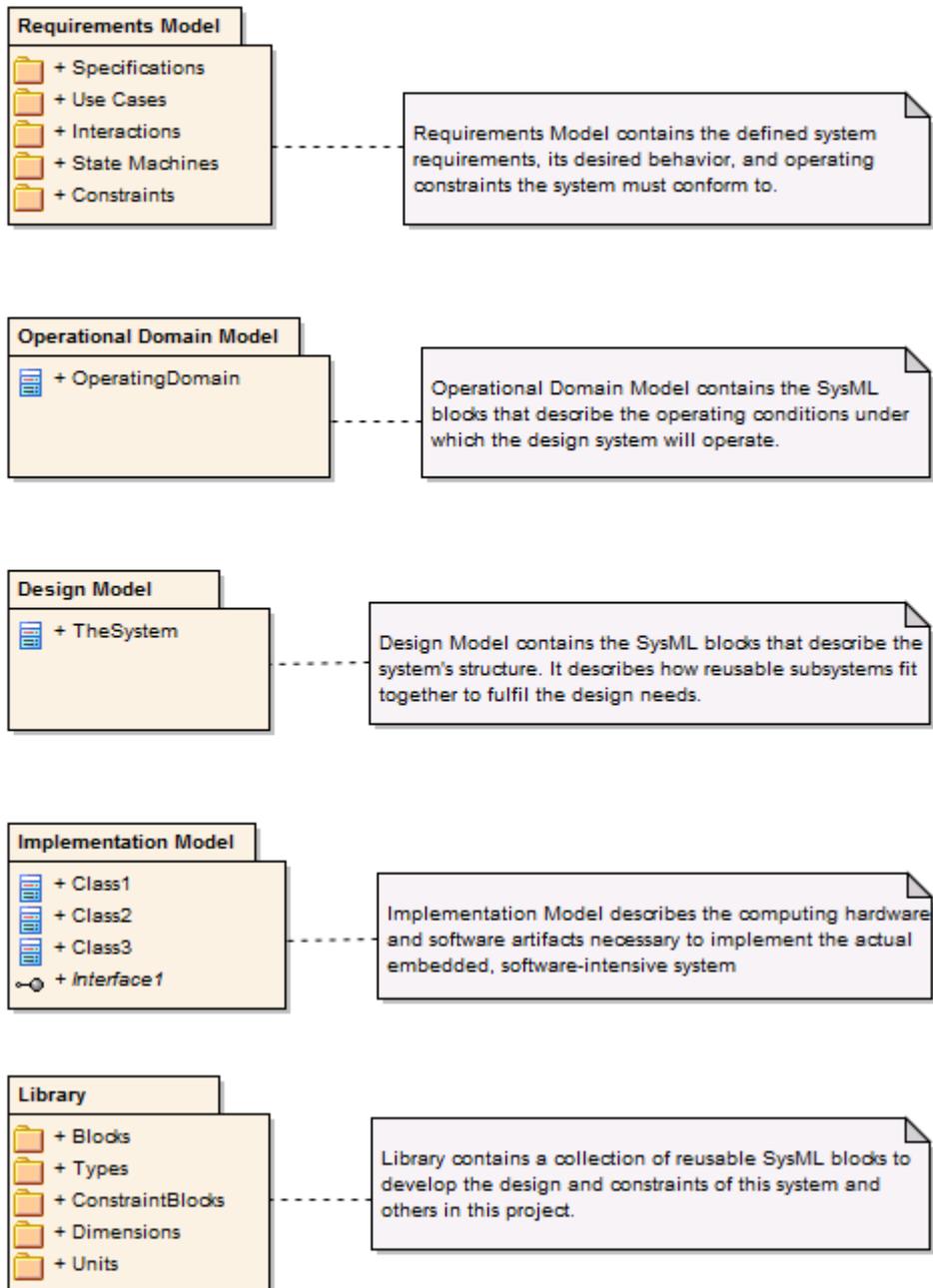
Example:

The following model structure is created in the **Project Browser**:



The *Systems Engineering Model* diagram, shown below, encapsulates the key components of the Systems Engineering model.

Systems Engineering Model captures the requirements, operating conditions and performance constraints of a complex system. It captures the design of the system in terms of distinct, configurable, reusable subsystem components.



Notes:

- Systems Modeling Language (SysML) is available in the Systems Engineering edition and Ultimate edition of Enterprise Architect

12.1 SysML

Topics:

Topic	Detail	See also
Introduction	<p>The following text is derived from the official OMG SysML site of the Object Management Group</p> <p><i>The OMG Systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities.</i></p> <p><i>The language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models.</i></p> <p><i>SysML was developed in response to requirements developed jointly by the OMG and the International Council on Systems Engineering (INCOSE) by the diverse group of tool vendors, end users, academia, and government representatives.</i></p>	OMG SysML
SysML In Enterprise Architect	<p>Enterprise Architect enables you to develop SysML models quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer</p> <p>The SysML technology provides:</p> <ul style="list-style-type: none"> • Each of the nine SysML diagram types, accessed through the New Diagram dialog • A collection of SysML pages in the Toolbox that contain each of the supported SysML elements and relationships • SysML element and relationship entries in the Toolbox Shortcut Menu and Quick Linker • Simulation of SysML Parametric diagrams, which supports engineering analysis of critical system parameters including the evaluation of key metrics such as performance, reliability and other physical characteristics 	Add New Diagram ^[568] Toolbox Shortcut Menu ^[551] Quick Linker ^[622] Simulate a SysML Model ^[1570]
SysML Toolboxes	<p>You can access the SysML pages of the Toolbox through the More tools SysML 1.1 and SysML 1.2 menu options</p> <p>You can also enable SysML as the active technology to access the Toolbox pages directly</p> <p>The following sets of Toolbox pages are available:</p> <ul style="list-style-type: none"> • Model Elements contains the constructs required to build SysML models, package structures and views • Block Definition contains the constructs required to design SysML blocks, constraint blocks, data and value types • Internal Block contains the constructs required 	Model Elements ^[1561] Block Definition ^[1556] Internal Block ^[1559] Parametrics ^[1563] Activity ^[1553] Interaction ^[1558] State Machine ^[1565] Use Case ^[1567] Requirements ^[1564] Default Tools Toolbar ^[107]

Topic	Detail	See also
	<p>to design SysML block compositions within Internal Block Diagrams</p> <ul style="list-style-type: none"> • Parametrics contains the constructs required to construct SysML Parametric Diagrams using constraint blocks • Activity contains the constructs required to construct SysML Activity models • Interaction contains the constructs required to construct SysML interactions and Sequence diagrams • State Machine contains the constructs required to build SysML State Machines • Use Case contains the constructs required to build SysML Use Case models • Requirements contains the constructs required to build SysML Requirements models <p>With the Model Elements pages there is a set of SysML Common elements and relationships; these are also provided with the other SysML Toolbox pages if the active technology is set to SysML 1.1 or SysML 1.2 on the Default Tools toolbar</p>	
Convert SysML 1.1 Models to SysML 1.2	Enterprise Architect enables you to migrate a SysML 1.1 model (or part of a model) to SysML 1.2, using the Automation Interface	Migrate SysML 1.1 Model to SysML 1.2 ^[1578]
Disable SysML	If you prefer not to use SysML in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies)	MDG Technologies ^[1033]

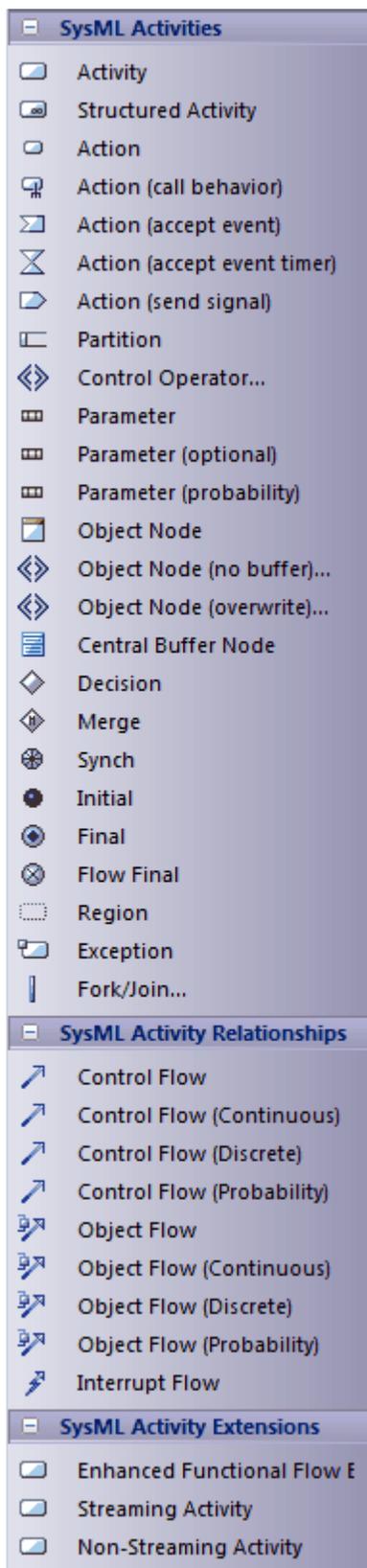
Notes:

- Systems Modeling Language (SysML) is supported in the Systems Engineering and Ultimate editions of Enterprise Architect

12.1.1 SysML Activity

These pages provide the SysML elements to construct SysML Activity models.

Access: [View | Diagram Toolbox: More Tools | SysML 1.2 | SysML 1.2 Activity](#)



Page	Item	Action
SysML Activities	Activity	Define a SysML Block of executable behavior as a UML Activity

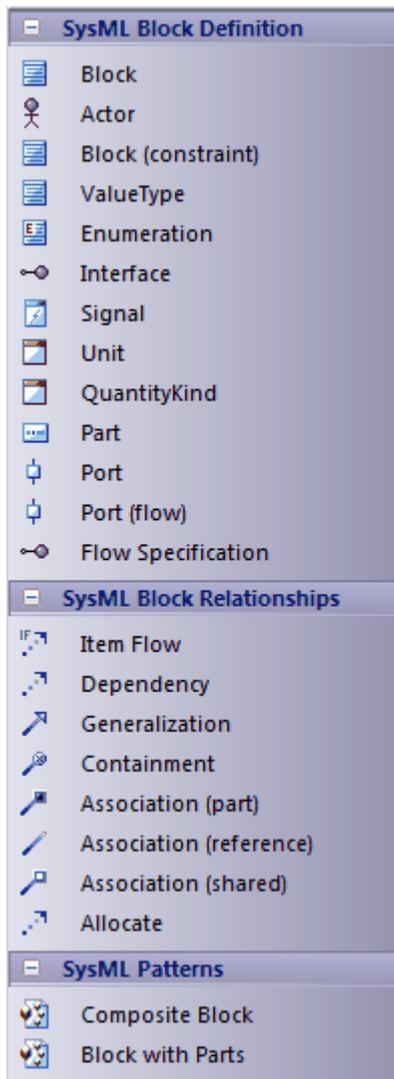
Page	Item	Action
	Structured Activity	Define a SysML Block of executable behavior as a UML Structured Activity
	Action	Declare a unit of execution in an Activity as a UML Action.
	Action (call behavior)	Declare a unit of execution that calls another behavior
	Action (accept event)	Declare a unit of execution that accepts an event raised by the system
	Action (accept event timer)	Declare a unit of execution that accepts an event raised by a time epoch
	Action (send signal)	Declare a unit of execution that sends a signal as an event
	Partition	Create an Activity Partition to group execution elements together by the node responsible for their execution
	Control Operator	Control the execution of an Activity
	Parameter	Provide access to input and output objects within the Activity
	Parameter (optional)	Define a parameter whose contents are optional in the Activity's execution
	Parameter (probability)	Tag a parameter with probability of the likelihood of the parameter's use in the Activity
	Object Node	Declare a variable in the Activity, typed by a ValueType, DataType or Block
	Object Node (no buffer)	Declare an ObjectNode in an Activity which discards unconsumed tokens
	Object Node (overwrite)	Declare an ObjectNode in an Activity which overwrites tokens
	Central Buffer Node	Declare an ObjectNode that stores tokens for consumption throughout the Activity
	Decision	Create a branch of control in an Activity based on a decision
	Merge	Merge two or more Activity control branches together
	Synch	Establish a rendezvous point for two or more Activity flows, in order to synchronize their execution in the Activity
	Initial	Declare the start of Activity's execution
	Final	Declare the end of an Activity's execution, and the termination of the Activity
	Flow Final	Declare the end of an Activity's execution path without terminating the Activity
	Region	Group a subset of an Activity into a common execution context
	Exception	Declare a node of execution that happens outside the normal flow of execution of an Activity
	Fork/Join	Simultaneously branch / join a set of Control or Object Flows

Page	Item	Action
SysML Activity Relationships	Control Flow	Establish a flow of logic between two Activity nodes
	Control Flow (Continuous)	Declare a continuous control flow
	Control Flow (Discrete)	Declare a discrete control flow
	Control Flow (Probability)	Tag a control flow with a probability of the likelihood of the flow's traversal
	Object Flow	Establish a flow of objects (data) between two Activity nodes
	Object Flow (Continuous)	Declare a continuous object flow
	Object Flow (Discrete)	Declare a discrete object flow
	Object Flow (Probability)	Tag an object flow with a probability of the likelihood of the flow's traversal
	Interrupt Flow	Declare a control flow that interrupts flow within a Region
SysML Activity Extensions	Enhanced Functional Flow Block Diagrams	Declare an Activity used to contain an Enhanced Functional Flow Block Diagram (EFFBD)
	Streaming Activity	Declare an Activity where the flow of tokens passes through its parameters continuously throughout the Activity's execution
	Non-Streaming Activity	Declare an Activity where the flow of tokens passes through its parameters at the start of the Activity's execution

12.1.2 SysML Block Definition

These pages provide the SysML elements to design SysML blocks, constraint blocks, data and value types.

Access: [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Block Definition](#)



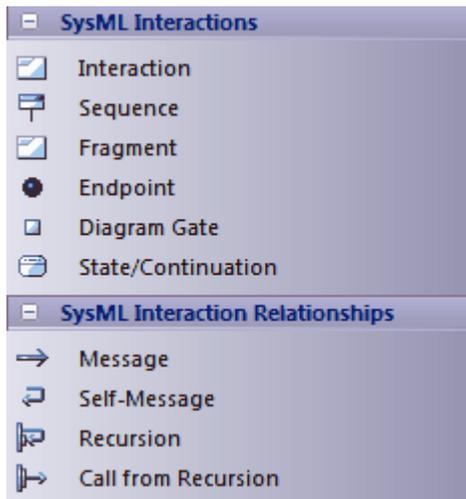
Page	Item	Action
SysML Block Definition	Block	Define a composite system entity in SysML
	Actor	Represent a user that interacts with one or more SysML systems
	Block (constraint)	Define a composite constraint as a system of parametric equations
	ValueType	Define a SysML quantity, expressed as a measurable dimension with specific units
	Enumeration	Define a data type as a set of symbols or values
	Interface	Define an element that describes a specification of an interaction point with properties and methods
	Signal	Define a SysML message, containing attributes, exchanged between system blocks in an interaction
	Unit	Represent a standard unit of measure in SysML

Page	Item	Action
	QuantityKind	Identify a measurable quantity in SysML
	Part	Describe the decomposition of a SysML Block subsystem in the context of its whole using instances of reusable SysML Blocks
	Port	Describe a structural interaction point of a SysML Block which, in turn, connects between interacting parts of a block
	Port (flow)	Describe what flows in and out of interacting SysML Blocks
	Flow Specification	Define a set of flow properties that correspond to individual pieces of a common interaction point
SysML Block Relationships	Item Flow	Specify the items that flow across a connector in an interaction point
	Dependency	Establish a traceable relationship describing how one element is dependant upon another
	Generalization	Describe an element as a specialized descendant of another element, containing additional properties and behavior
	Containment	Graphically display ownership of one element within a parent element
	Association (part)	Describe the characteristics of a connection between a SysML Block and its internal parts, such as the multiplicity and type
	Association (reference)	Describe the characteristics of a connection between separate SysML Blocks, such as the multiplicity and type
	Association (shared)	Describe the characteristics of a common connection between SysML Blocks, such as the multiplicity and type
	Allocate	Relate model elements together to formalize a refinement of behavior, structure, constraints or design expectations
SysML Patterns	Composite Block	A pattern that creates a Composite Block made up of Blocks related by Aggregation relationships
	Block with Parts	A pattern that creates a Composite Block made up of nested parts

12.1.3 SysML Interaction

These pages provide the SysML elements to construct SysML interactions and Sequence diagrams.

Access: [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML1.2 Interaction](#)

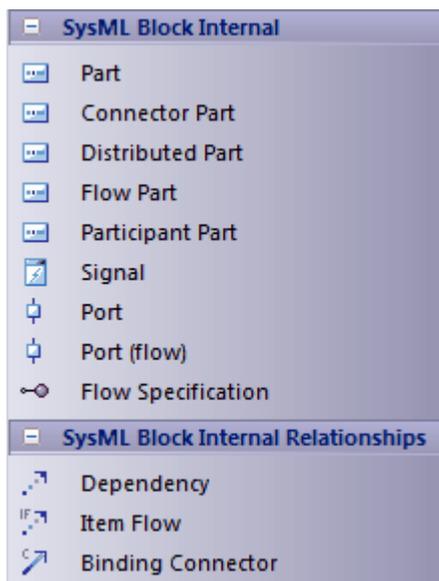


Page	Item	Action
SysML Interactions	Interaction	Define a SysML Block of executable behavior as a UML Interaction
	Sequence	Reference an instance of a SysML Block as a Lifeline in the Interaction
	Fragment	Declare a portion of an interaction as a group with specific behavior semantics
	Endpoint	Create an exit point for the Interaction
	Diagram Gate	Create an endpoint for the interaction, which bridges between nested interactions
	State/Continuation	Constrain the Interaction with assertions of the state that the lifeline is expected to be in
SysML Interactions Relationships	Message	Describe a message exchange between two lifelines in an Interaction
	Self-Message	Describe a message exchange between a lifeline and itself in an Interaction
	Recursion	Describe a recursive message exchange between a lifeline and itself in an Interaction
	Call from Recursion	Describe a message exchange between two lifelines within a recursive exchange

12.1.4 SysML Internal Block

These pages provide the SysML elements to design SysML block compositions within Internal Block Diagrams.

Access: [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Internal Block](#)

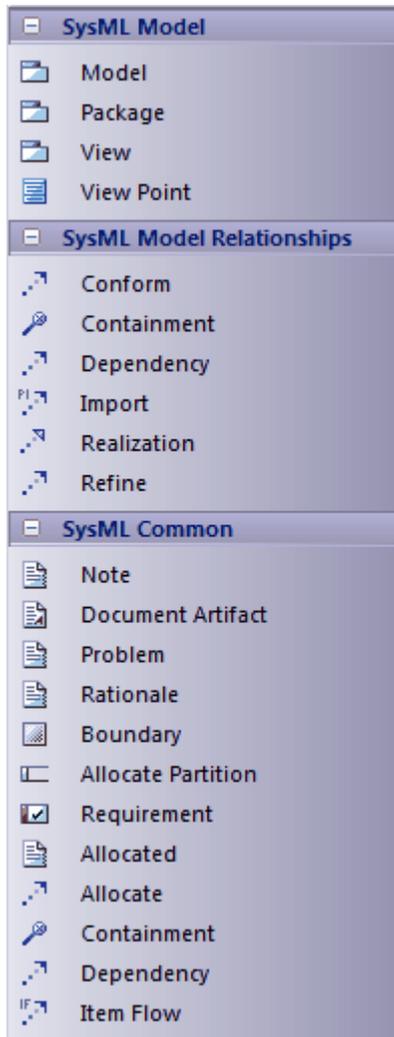


Page	Item	Action
SysML Block Internal	Part	Describe the decomposition of a SysML Block subsystem in the context of its whole, using instances of reusable SysML Blocks
	Connector Part	Create a SysML connector part
	Distributed Part	Create a SysML distributed part
	Flow Part	Create a SysML flow part
	Participant Part	Create a SysML participant part
	Signal	Define a SysML message, containing attributes, exchanged between system blocks in an interaction
	Port	Describe a structural interaction point of a SysML Block which, in turn, connects between interacting parts of a block
	Port (flow)	Describe what flows in and out of interacting SysML Blocks
	Flow Specification	Define a set of flow properties that correspond to individual pieces of a common interaction point
SysML Block Internal Relationships	Dependency	Establish a traceable relationship describing how one element is dependant upon another
	Item Flow	Specify the items that flow across a connector in an interaction point
	Binding Connector	Establish a connection between two parts in a system decomposition

12.1.5 SysML Model Elements

These pages provide the SysML elements to build SysML models, package structures and views.

Access: [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Model Elements](#)



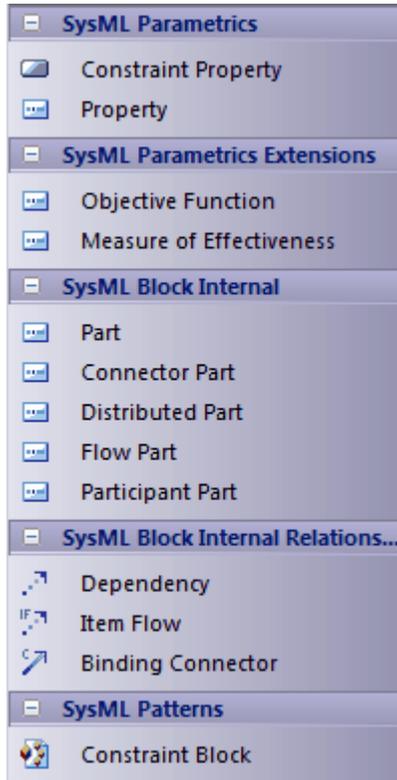
Page	Item	Action
SysML Model	Model	Create a Package containing a SysML Model
	Package	Group model constructs together in a single unit of containment
	View	Create a stereotyped Package that defines a SysML View of a system, from the perspective of a SysML View Point
	View Point	Create a stereotyped Class that defines a SysML View Point, which specifies the rules and conventions for the construction and use of Views
SysML Model Relationships	Conform	Establish a conformance dependency of a View to the defining View Point

Page	Item	Action
	Containment	Graphically display ownership of one element within a parent one
	Dependency	Establish a traceable relationship describing how one element is dependant upon another
	Import	Represent a reuse of elements from one model package in another
	Realization	Identify a design fulfillment of a specification between elements
	Refine	Represent a refinement of one element by another
SysML Common	Note	Create a textual annotation that can be attached to a set of elements of any other type The attachment is created separately, using a Notelink connector
	Document Artifact	Attach a linked document to the diagram by associating this element with an RTF document
	Problem	Document the failure of model elements to satisfy a requirement
	Rationale	Document the justification for decisions
	Boundary	Define a conceptual boundary, to visually group logically related elements
	Allocate Partition	Contains elements deemed to be allocated to the classifier of the partition
	Requirement	Specify the capabilities of the system, or the conditions that it should satisfy
	Allocated	The source element being allocated to the target element in an Allocate relationship
	Allocate	Relate model elements to formalize a refinement of behavior, structure, constraints or design expectations The Allocate relationship points from the element being allocated to the element that is the target of the allocation
	Containment	Graphically display ownership of one element within a parent one
	Dependency	Establish a traceable relationship describing how one element is dependant upon another
	Item Flow	Specify the items that flow across a connector in an interaction point.

12.1.6 SysML Parametrics

These pages provide the SysML elements to construct SysML Parametric Diagrams using constraint blocks.

Access: [View | Diagram Toolbox: More Tools | SysML 1.2 | SysML 1.2 Parametrics](#)



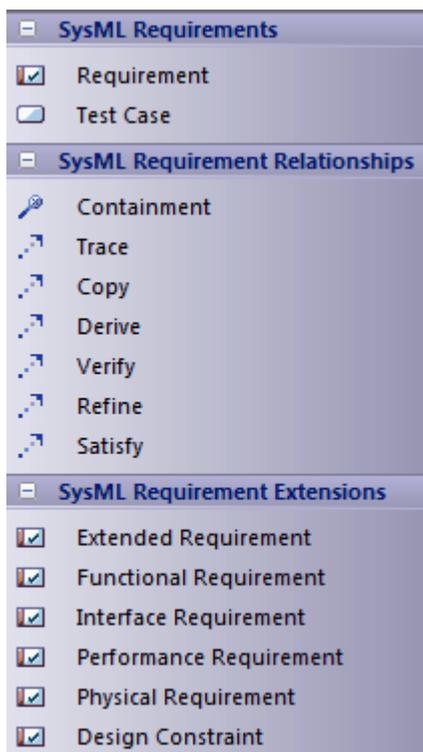
Page	Item	Action
SysML Parametrics	Constraint Property	Instantiate a Constraint Block for use in a Parametric diagram
	Property	Define a SysML property typed by a DataType, ValueType or Block
SysML Parametrics Extensions	Objective Function	Define a SysML Constraint Block for use as an objective function to evaluate Measures of Effectiveness (MOEs)
	Measure of Effectiveness	Define a SysML property for use as a Measure of Effectiveness (MOE)
SysML Block Internal	Part	Describe the decomposition of a SysML Block subsystem in the context of its whole, using instances of reusable SysML Blocks
	Connector Part	Create a SysML connector Part
	Distributed Part	Create a SysML distributed Part
	Flow Part	Create a SysML flow Part
	Participant Part	Create a SysML participant Part

Page	Item	Action
SysML Block Internal Relationships	Dependency	Establish a traceable relationship describing how one element is dependant upon another
	Item Flow	Specify the items that flow across a connector in an interaction point
	Binding Connector	Establish a connection between two Parts in a system decomposition
SysML Patterns	Constraint Block	A pattern that creates a typical Constraint property that owns two Parts

12.1.7 SysML Requirements

These pages provide the SysML elements to build SysML Requirements models.

Access: [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.1 Requirements](#)



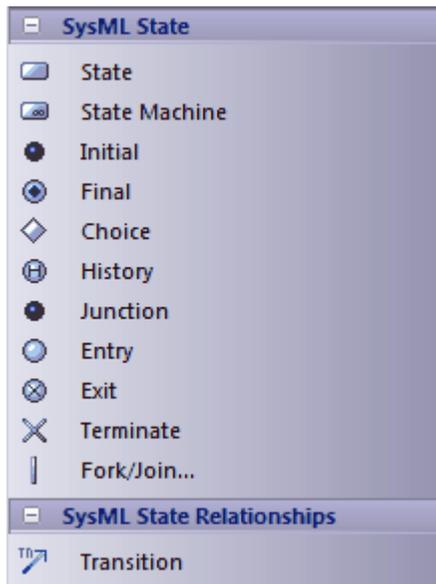
Page	Item	Action
SysML Requirements	Requirement	Specify the capabilities of the system, or the conditions that it should satisfy
	Test Case	Describe the verification of a Requirement through methods of inspection, analysis, demonstration or testing
SysML Requirement	Containment	Graphically display ownership of one element within a parent element

Page	Item	Action
	Trace	Declare a trace relationship between a SysML Requirement and another SysML element
	Copy	Declare a copy of one SysML Requirement by another
	Derive	Derive a SysML Requirement from another
	Verify	Declare a verification of a SysML Requirement by another SysML element
	Refine	Declare a refinement of a SysML Requirement by another SysML element
	Satisfy	Declare that the SysML Requirement is satisfied by another SysML element
SysML Requirement Extensions	Extended Requirement	Extend a SysML Requirement with additional Tag properties
	Functional Requirement	Declare a SysML Requirement that describes the operation, or behavior, that the system must perform
	Interface Requirement	Declare a SysML Requirement that describes how the system connects, or interfaces with, other systems
	Performance Requirement	Declare a SysML Requirement that describes how the system performs against defined capabilities or conditions
	Physical Requirement	Declare a SysML Requirement that describes the physical characteristics, or physical constraints, of the system
	Design Requirement	Declare a SysML Requirement that specifies a constraint on the implementation of the system

12.1.8 SysML State Machine

These pages provide the SysML elements to build SysML State Machines.

Access: [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 State Machine](#)

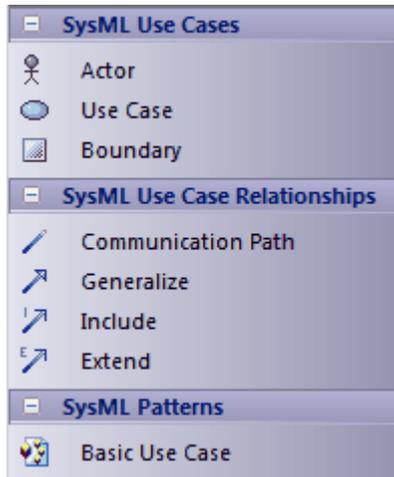


Page	Item	Action
SysML State	State	Declare a significant condition in the life of a SysML Block within its State Machine
	State Machine	Describe the life-cycle behavior of a SysML Block in terms of its states and transitions
	Initial	Declare the starting state of the State Machine
	Final	Declare the ending state of the State Machine, and its completion
	Choice	Declare a Junction with a mandatory 'else' transition
	History	Represent the last active State of the State Machine prior to its interruption
	Junction	Declare a decision point at which a Transition branches out into multiple guarded, alternative paths
	Entry	Declare an Entry point between State Machines, Substate Machines and Regions
	Exit	Declare an Exit point between State Machines, Substate Machines and Regions
	Terminate	Declare a termination State in which the State Machine no longer operates
	Fork/Join	Simultaneously branch / join a set of Transitions
SysML State Relationships	Transition	Establish a life-cycle path between one State and another, based on its operational conditions

12.1.9 SysML Use Case

These pages provide the SysML elements to build SysML Use Case models.

Access: [View](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.2](#) | [SysML 1.2 Use Cases](#)



Page	Item	Action
SysML Use Cases	Actor	Represent a user that interacts with one or more SysML systems
	Use Case	Describe the expected functionality of a system as a UML Use Case
	Boundary	Graphically bound elements in a diagram with a border
SysML Use Case Relationships	Communication Path	Declare which Actors perform in the Use Case
	Generalize	Describe an element as a specialized descendant of another element, containing additional properties and behavior
	Include	Describe one Use Case as a subset of another
	Extend	Describe one Use Case as an extension of another
SysML Patterns	Basic Use Case	A pattern that creates a typical simple Use Case diagram of Actor, Use Case and System Boundary

12.2 SysML Parametric Models

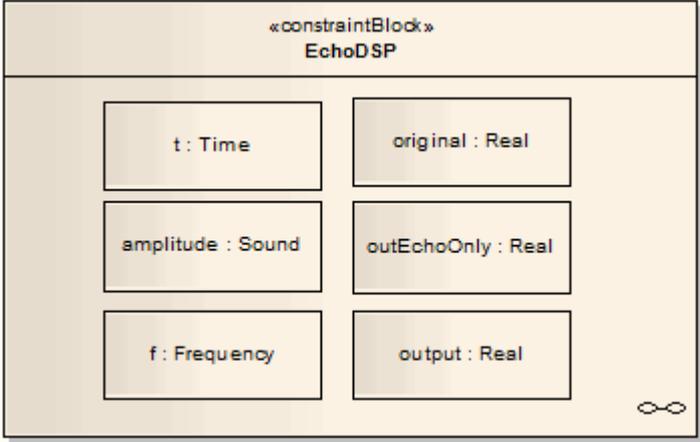
SysML Parametric models support the engineering analysis of critical system parameters, including the evaluation of key metrics such as performance, reliability and other physical characteristics.

Topics:

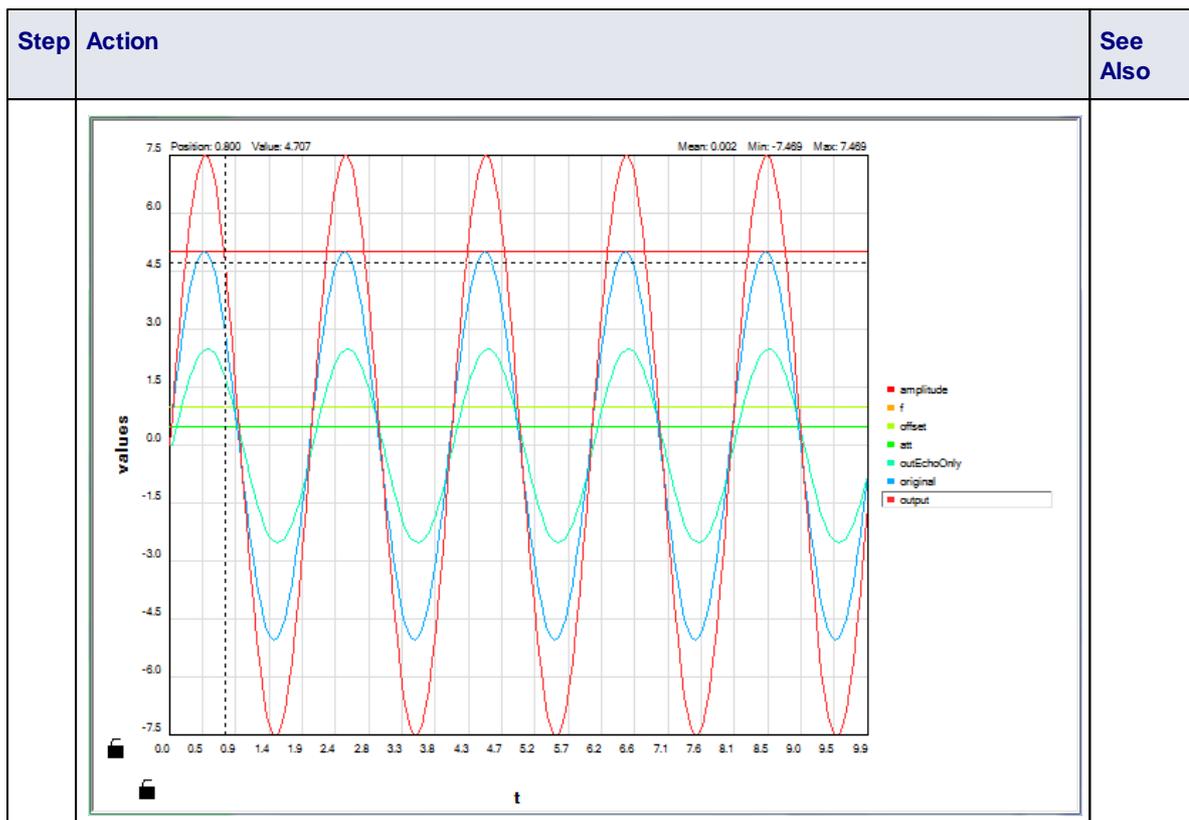
Topic	Detail	See also
SysML Parametric Models - Concept	<p>SysML Parametric Models unite requirements models with system design models by capturing executable constraints based on complex mathematical relationships</p> <p>The following text is derived from the SysML entry in the online Wikipedia</p> <p><i>The advantages of SysML over UML for systems engineering become obvious if you consider a concrete example, such as modeling an automotive system. With SysML you can use Parametric diagrams to precisely define performance and mechanical constraints such as maximum acceleration, curb weight, air conditioning capacity, and interior cabin noise management.</i></p> <p>For further information on the concepts of SysML Parametric models, refer to the official OMG SysML website and its linked sources</p> <p>Enterprise Architect enables you to develop SysML Parametric models quickly and simply; these models can also be simulated</p>	<p>SysML (Online Resource)</p> <p>OMG SysML (Online Resource)</p>

How To:

To create a Parametric model, follow the steps below:

Step	Action	See Also
1	<p>Create a collection of <i>SysML Constraint Blocks</i> that formally describe the function of a constraint in a simulation model</p> <p>Each Constraint Block contains properties that describe its input and output parameters, as well as <i>Element Script</i> that describes the constraint's executable component</p> 	
2	Right-click on each of the <i>constraintBlocks</i> and select the SysML Add Element Script	

Step	Action	See Also
	<p>context menu option to add script to the constraint block</p> <p>This is where you express the relationship / behavior of the constraint block as an executable script</p>	
<p>3</p>	<p>Create a SysML Constraint Block to contain the Parametric model to simulate; the Parametric model contains properties and occurrences of constraint blocks as Constraint Property elements, connected in a Parametric Diagram</p>	
<p>4</p>	<p>Right-click within a Parametric Diagram and select the SysML Simulate Diagram... context menu option</p>	
<p>5</p>	<p>Depending on your configuration selections, the simulation's results are either written to a comma-separated CSV file or graphed in a 2-dimensional plot</p>	

**Notes:**

- Systems Modeling Language (SysML) is supported in the Systems Engineering and Ultimate editions of Enterprise Architect

Learn More:

- [SysML](#)^[1552]
- [Simulating a SysML Model](#)^[1570]

12.2.1 Simulate a SysML Model

You simulate a SysML model from a Parametric diagram, using the Simulation Configuration dialog.

Access: **Parametric Diagram Context Menu | SysML | Simulate Diagram..**

How To:

To simulate a SysML model, follow the steps below:

Step	Action	See Also
1	<p>The Parameters panel lists all of the parameters that can be assigned input</p> <p>Select each of the required parameters and click on the right Arrow button to assign them as input</p> <p>Parameters designated as input parameters are listed in the Inputs panel on the right.</p>	

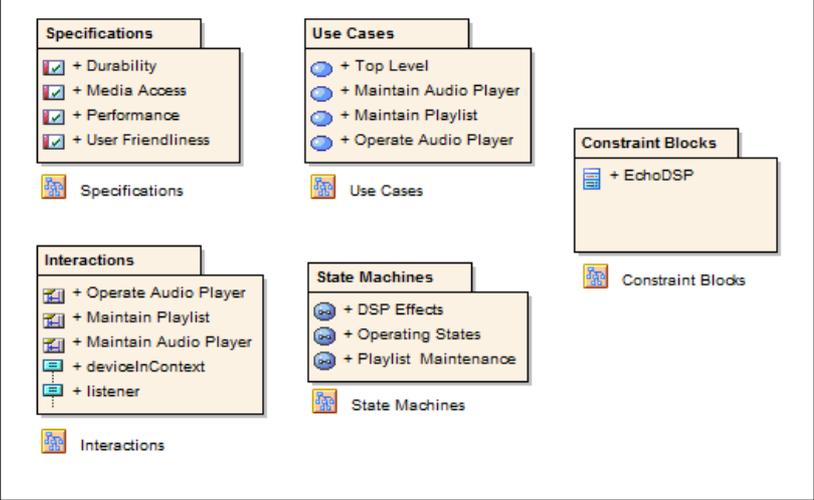
Step	Action	See Also
	There must be at least one input parameter assigned for the simulation to execute	
2	Assign a set of values for each of the designated input parameters For each input parameter, in the Input Values panel select one of the two possible value kinds: <ul style="list-style-type: none"> • Discrete - To enter a constant or a comma-separated range of discrete values • Range - To enter a range of values beginning at the From value and ending at the To value; the input values are incremented by the Step value 	
3	Specify the classes of output value: <ul style="list-style-type: none"> • Parameters - To output the parameters' data, select the checkbox • Variables - To output the data generated within each internal variable, select the checkbox; internal variables are automatically generated by the simulator 	
4	Specify how the simulation results are to be reported The Output Format panel enables you to choose how the simulation outputs the simulation data: <ul style="list-style-type: none"> • Plot To Graph: To plot the results on a 2-dimensional graph, select the checkbox; if you select this option, you must specify an input parameter for the plot's X Axis • Title - To enter a title for the graph, type in the title text • Output to File - To output the results to a CSV text file, select the checkbox and type or browse (...) for the file name 	
5	Click on the OK button to execute the simulation	

Notes:

- Systems Modeling Language (SysML) is supported in the Systems Engineering and Ultimate editions of Enterprise Architect

12.3 Create a Requirements Model

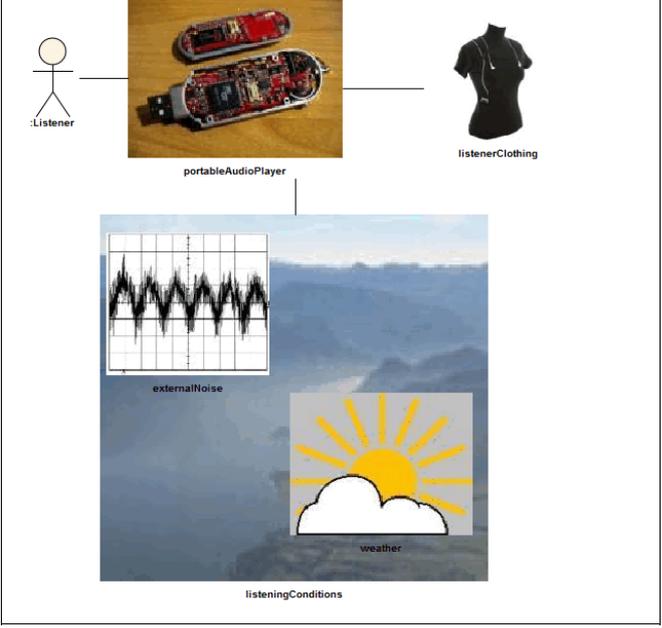
Topics:

Topic	Detail	See also
<p>Example</p>	<p>The SysML <i>Requirements Model</i> provides the system requirements, the expected abstract behavior and the operating constraints that the designed system must conform to</p> <p>The following diagram shows an example requirements model for a <i>Portable Audio Player</i></p>  <p>In the example Requirements Model, each of the child packages contains child models that capture the following aspects of the system's requirements:</p> <ul style="list-style-type: none"> • The <i>Specifications</i> package contains SysML Requirements describing the overall expectations of the designed system • The <i>Use Cases</i> package contains SysML Use Cases that describe the general interaction between the system and its users • The <i>Interactions</i> package contains SysML Interactions that describe a detailed sequence of interactions between the system and its users • The <i>State Machines</i> package contains SysML State Machines that describe each of the operational states the designed system has • The <i>Constraint Blocks</i> package contains SysML ConstraintBlocks that describe the expected performance and operating boundaries of the system 	

12.4 Create an Operational Domain Model

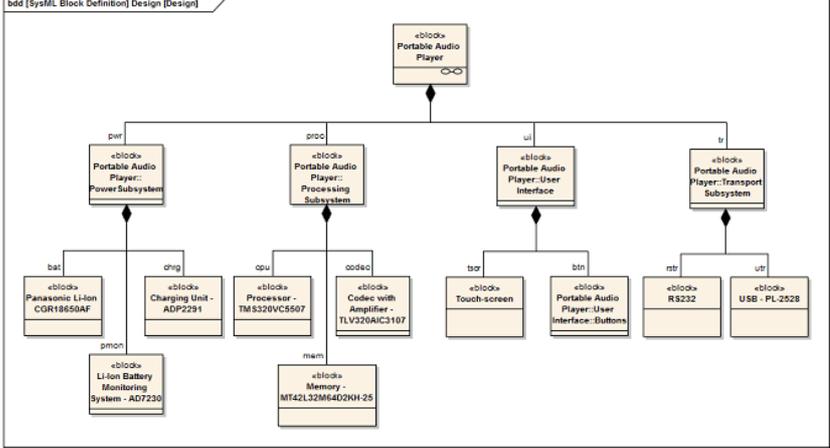
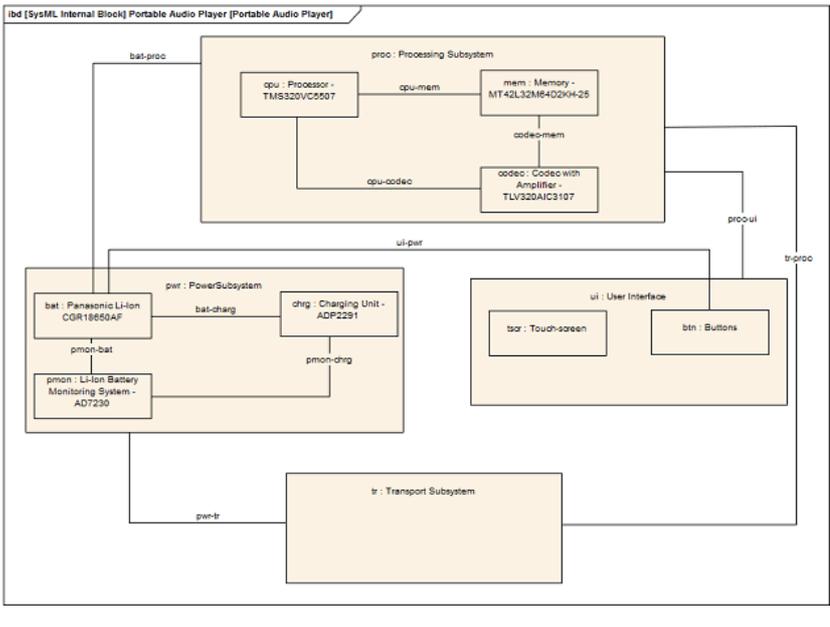
Topics:

Topic	Detail	See also
<p>Abstract</p>	<p>The SysML <i>Operational Domain Model</i> defines the system's operating environment, which describes the operating conditions that the system is intended to operate within</p> <p>The following diagram shows an example Operational Domain model for a <i>Portable Audio Player</i>; the SysML Block Definition Diagram describes the Operational Domain (in this example - the <i>ListeningDomain</i>) as a system composition</p> <div data-bbox="507 752 1166 1249" data-label="Diagram"> <pre> graph TD subgraph "bdd [SysML Block Definition] Listening Domain [Listening Domain]" LPD[«block» ListeningDomain] LAP[«block» Portable Audio Player] C[«block» Clothing] E[«block» Environment] EN[«block» Environment: Noise] EW[«block» Environment: Weather] L[Listener] LPD --> LAP LPD --> C LPD --> E E --> EN E --> EW L -.- LAP end </pre> </div> <p>In this example, the <i>ListeningDomain</i> is defined as a system containing other subsystems within it; the domain contains subsystems that define the Listener (i.e. User), the Portable Audio Player, Clothing (which the user wears), and the External Environment</p> <p>Details of the <i>ListeningDomain</i> system are further detailed in the <i>ListeningDomain</i>'s Internal Block Diagram:</p>	

Topic	Detail	See also
	<p data-bbox="507 320 850 342">ibd [SysML Internal Block] ListeningDomain [ListeningDomain]</p>  <p>The diagram shows an internal block definition for 'ListeningDomain'. It contains four components: a stick figure labeled ':Listener', a photograph of a portable audio player labeled 'portableAudioPlayer', a black t-shirt labeled 'listenerClothing', and a landscape image with a sun and clouds labeled 'listeningConditions'. The 'listeningConditions' block contains a waveform labeled 'externalNoise' and a sun icon labeled 'weather'. Lines connect the 'Listener' to the 'portableAudioPlayer', and the 'portableAudioPlayer' to the 'listenerClothing' and 'listeningConditions' blocks.</p> <p data-bbox="507 1003 1169 1167">In this example, the <i>ListeningDomain's</i> system's detailed composition shows how the Portable Audio Player and other sub-systems fit together to form the Listening Domain; it also describes the binding relationships between the parts, which describe how the parts are functionally bound to one another</p>	

12.5 Compose System Design

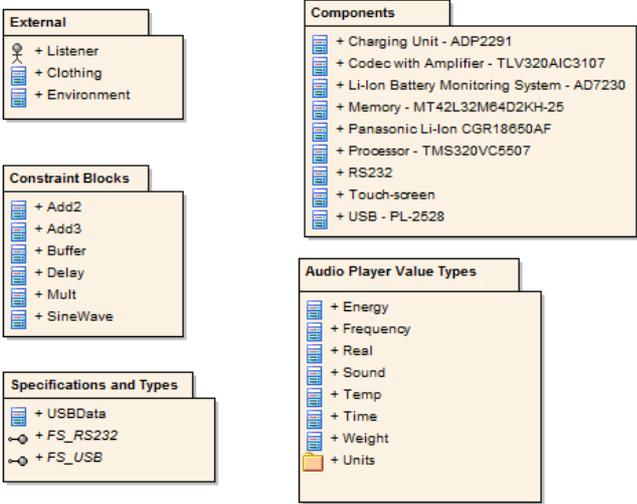
Topics:

Topic	Detail	See also
<p>SysML Design Model</p>	<p>The <i>SysML Design Model</i> contains the blocks that define the system's composition; it describes the manner in which reusable subsystems fit together to fulfill the design requirements</p> <p>The following diagram shows an example Design Model for a <i>Portable Audio Player</i>; the <i>SysML Block Definition Diagram</i> describes the <i>Portable Audio Player</i> as a composition of various reusable off-the-shelf subsystems and in-house designed ones</p>  <p>The diagram is a SysML Block Definition Diagram (bdd) titled "[SysML Block Definition] Design [Design]". It shows a hierarchical structure of blocks. At the top is the "Portable Audio Player" block. Below it are four main subsystems: "Power Subsystem", "Processing Subsystem", "User Interface", and "Transport Subsystem". Each subsystem is further decomposed into specific hardware and software components. For example, the Power Subsystem includes a battery, charging unit, and battery monitoring system. The Processing Subsystem includes a processor, memory, and audio codec. The User Interface includes a touch-screen and buttons. The Transport Subsystem includes RS232 and USB ports.</p> <p>In the example above, the <i>Portable Audio Player</i> is defined as a SysML system containing subsystems that perform specific tasks; the design contains subsystems for supplying power, performing playback and audio processing, interfacing with other devices and the user interface</p> <p>Details of the <i>Portable Audio Player's</i> composition are further described in detail within the <i>Portable Audio Player's Internal Block Diagram</i>:</p>  <p>The diagram is a SysML Internal Block Diagram (ibd) titled "[SysML Internal Block] Portable Audio Player [Portable Audio Player]". It shows the internal composition of the Portable Audio Player. The main block is the "Processing Subsystem", which contains a "Processor", "Memory", and "Audio Codec". The "Power Subsystem" contains a "Battery", "Charging Unit", and "Battery Monitoring System". The "User Interface" contains a "Touch-screen" and "Buttons". The "Transport Subsystem" is also shown. The diagram illustrates how these subsystems are interconnected and how they share resources like memory and data.</p>	

Topic	Detail	See also
	<p>In this example, the <i>Portable Audio Player's</i> composition is described, detailing how each of the sub-systems is structured</p> <p>It also describes the binding relationships between the parts, which describe how they are functionally bound to one another; for example, the CPU, Memory and Codec are interfaced together in the Processing Subsystem</p>	

12.6 Create Reusable Subsystems

Topics:

Topic	Detail	See also
<p>SysML Design Model</p>	<p>Model Based Systems Engineering provides the flexibility and expressiveness to define complex systems quickly and effectively, by reusing common entities across design projects</p> <p>A <i>Library</i> is a package containing many reusable subsystems, parametric constraints, common data types and common value types, dimensions and units</p> <p>The following diagram shows an example library model:</p>  <p>In the example Library, each of the child packages contains child models that capture the following reusable entities:</p> <ul style="list-style-type: none"> • <i>Blocks</i> defining systems such as those listed in the <i>Components</i> package, or those defined in the <i>External</i> package • <i>ConstraintBlocks</i> defining parametric constraints for use in parametric models. • <i>Value Types</i> describing quantities, expressed as measurable dimensions in specific units • <i>Data Types</i> and <i>Flow Specifications</i> describing data structures and <i>Flows</i> 	

12.7 Migrate SysML 1.1 Model to SysML 1.2

Enterprise Architect enables you to migrate a SysML 1.1 model (or part of a model) to SysML 1.2, using the Automation Interface function *Migrate()*. This function updates the Tagged Values and, if required, stereotypes to SysML1.2 for all elements, attributes, connectors and diagrams under the selected package or element.

Example:

The following VB script calls the *Migrate()* function to migrate the package or element to SysML 1.2:

```
Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "SysML1.1", "SysML1.2"

    'refresh the model
    If lngPackageID <> 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub

Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package
    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed", 0, "SysML Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Package Migration Completed", 0, "SysML Migration"
    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "SysML Migration"
    End If
End Sub

Sub Main
    MigrateSelectedItem
End Sub

Main
```

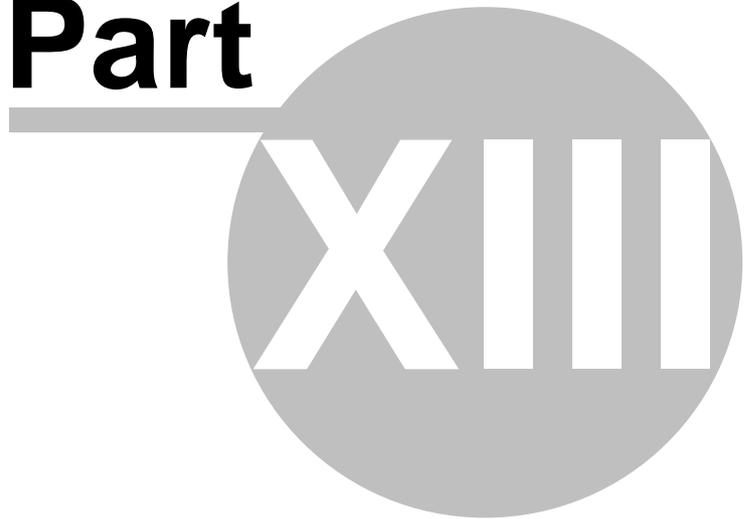
Notes:

- The stereotype *dimension* is changed to *quantitykind* when migrating from SysML1.1 to SysML1.2
- The stereotype *dataType* is removed from SysML1.2
- The Tagged Value *dimension* is migrated to *quantitykind*; this applies to stereotypes *unit* and *valueType*
- The Tagged Value *isConjugated* of stereotype *flowport* is migrated to custom properties

Learn More:

- [Migrate\(\)](#)^[1942]

Part



13 SOA and XML



This section explains how to import and generate:

- XSD
- WSDL

It also describes how to generate and export *Meta-Object* Facility (**MOF**) models, along with describing **SoaML** and **SOME**.

Learn More:

- [XML Schema](#)^[1581]
- [Web Services - WSDL](#)^[1611]
- [SoaML](#)^[1624]
- [SOME](#)^[1628]
- [Generating MOF](#)^[1630]

13.1 XML Schema - XSD

XML Schema Definition (XSD), also known as **XML Schema**, is a **W3C** XML technology that is used to specify the rules to which an XML document must adhere.

Enterprise Architect enables rapid modeling, forward engineering and reverse engineering of XML Schema. XSD support is critical for the development of a complete *Service Oriented Architecture* (SOA), and the coupling of UML 2.3 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization.

Learn More:

- [W3C XML Schema](#) (Online Resource)
- [Model XSD using Toolbox](#)^[1581]
- [Abstract XSD Models](#)^[1595]
- [Generate XSD](#)^[1598]
- [Import XSD](#)^[1600]
- [UML Profile for XSD](#)^[1602]

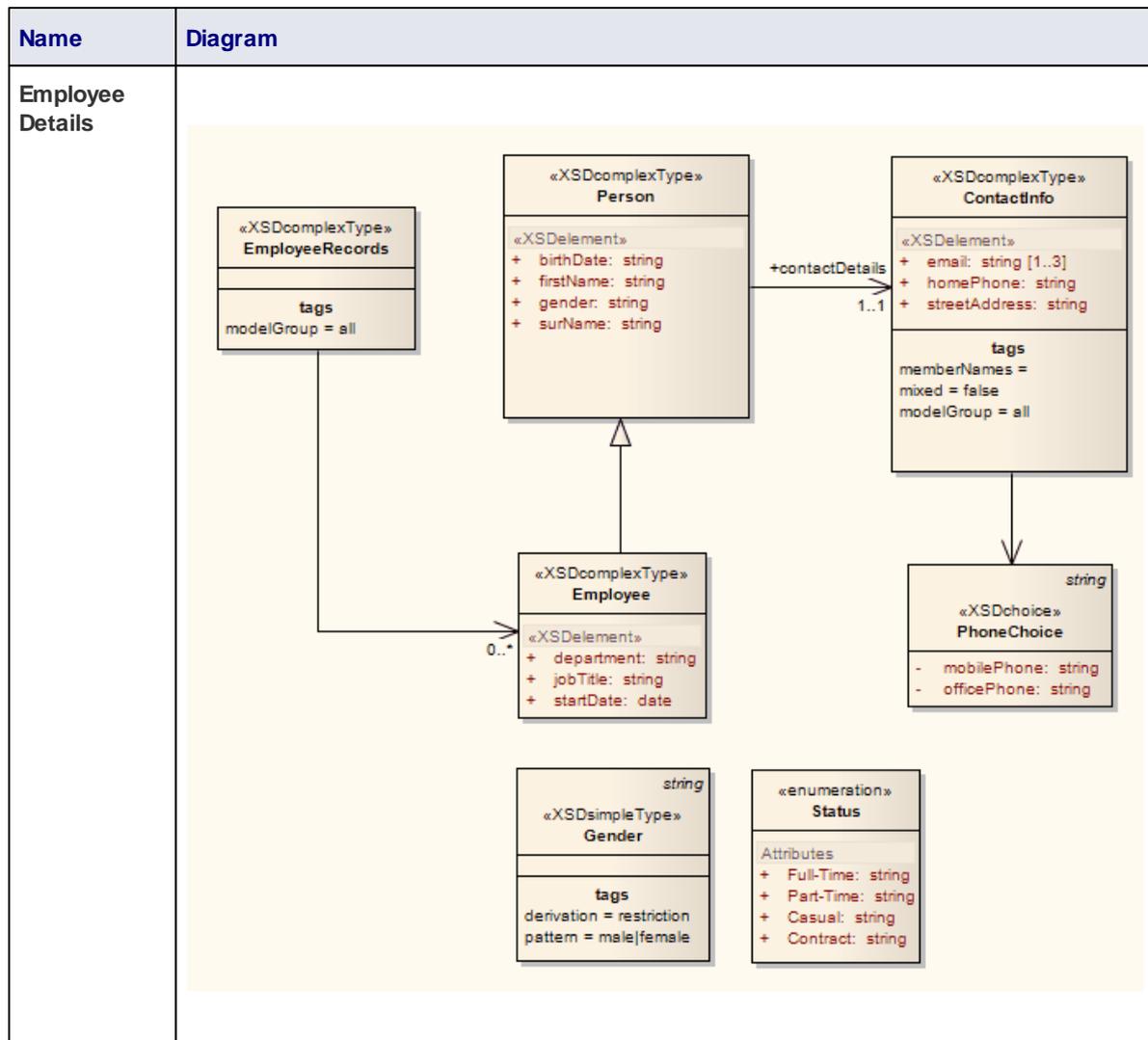
13.1.1 Model XSD

XML schemas are modeled using UML **Class** diagrams. The **XML Schema** page of the **Toolbox** provides built-in support for the **UML profile for XSD**. This enables an abstract UML Class model to be automatically generated as a **W3C XML Schema** (XSD) file.

Access: [View | Toolbox > More tools | XML Schema](#)

How To:

Step	Action	See Also
1	Create a package in the Project Browser	
2	Click on the New Diagram icon in the Project Browser	Project Browser Toolbar ^[452]
3	Select UML Structural in the Select From field and Class in the Diagram Types field	
4	Click on the More tools option in the Toolbox and select XML Schema from the list of Toolboxes	Diagram Toolbox ^[546]
5	Click on the Schema icon from the Toolbox and drag it into the created Class diagram When the New Diagram dialog is displayed, Select UML Structural in the Select From field and Class in the Diagram Types field to create a Class diagram.	Schema Icon ^[1583]
6	Enterprise Architect creates a new <i>XSDschema</i> stereotyped package with a Class diagram under it. Open this Class diagram and use the constructs from the XML Schema Toolbox to model the XML Schema Alternative: XML Schemas can also be modeled using the UML Profile for XSD .	XML Schema Group ^[566] UML profile for XSD ^[1602]

Example:**Notes:**

- The UML attributes of the Classes map directly to XML elements or attributes
- Classes have no methods since there is no meaningful correspondence between Class methods and XSD constructs

Modeling Restrictions:

The following XML Schema constructs cannot be modeled in Enterprise Architect:

- *appinfo*
- *field*
- *key*
- *keyref*
- *notation*
- *redefine*
- *selector*
- *substitutionGroup*

- *unique*

13.1.1.1 Schema Package

The **Schema** icon in the **XML Schema** Toolbox creates an *XSDschema* stereotyped package that acts as a container for the XSD constructs, from which XML Schema can be generated.

XSD Properties:

Double-click on the *XSDschema* stereotyped package in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also																		
Schema Name	Specify the name for the Schema package																			
Target Namespace	Specify the namespace for this Schema package (<i>Optional</i>)																			
Prefix	Specify the abbreviated value that represents the Target Namespace (<i>Optional</i>)																			
Default Namespace	Specify the namespace for all non-prefixed <i>XSDelements</i> and <i>XSDattributes</i> (<i>Optional</i>)																			
Schema File	Specify the file path where the XML Schema for this package is to be generated.																			
XMLNS	<p>List specifying the additional namespace and namespace-prefix pairs used in this Schema package.</p> <table border="1"> <thead> <tr> <th>Button</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td></td> <td>Press the New button to add an entry in the list.</td> </tr> <tr> <td></td> <td>Double-click on an entry in the list to edit the values.</td> </tr> <tr> <th>Field</th> <th>Usage</th> </tr> <tr> <td>Prefix</td> <td>Specify the abbreviated value that represents t</td> </tr> <tr> <td>Namespace</td> <td>Specify the namespace</td> </tr> <tr> <td>OK</td> <td>Save the values entered and close the Names</td> </tr> <tr> <td>Cancel</td> <td>Discard the values entered and close the Nam</td> </tr> <tr> <td>Help</td> <td>Display this Help topic</td> </tr> </tbody> </table> <p>Press the Delete button to remove an entry from the list</p>	Button	Usage		Press the New button to add an entry in the list.		Double-click on an entry in the list to edit the values.	Field	Usage	Prefix	Specify the abbreviated value that represents t	Namespace	Specify the namespace	OK	Save the values entered and close the Names	Cancel	Discard the values entered and close the Nam	Help	Display this Help topic	
Button	Usage																			
	Press the New button to add an entry in the list.																			
	Double-click on an entry in the list to edit the values.																			
Field	Usage																			
Prefix	Specify the abbreviated value that represents t																			
Namespace	Specify the namespace																			
OK	Save the values entered and close the Names																			
Cancel	Discard the values entered and close the Nam																			
Help	Display this Help topic																			
OK	Save the values entered and close the dialog																			
Cancel	Discard the values entered and close the dialog																			
Help	Display this Help topic																			
UML	Open the UML Properties dialog	Properties																		

	This button is available when editing the Schema package.	
--	---	---

13.1.1.2 Global Element

The **Element** icon from the **XML Schema** Toolbox creates an *XSDtopLevelElement* stereotyped Class, which acts as a XSD global element, when dropped directly on a diagram.

XSD Properties:

Double-click on the *XSDtopLevelElement* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also						
Name	Specify the name for this global element							
Type	Specify either an XSD <i>built-in dataType</i> or <i>complexType</i> or <i>simpleType</i>							
Value	If Type is an XSD <i>built-in type</i> then optionally specify its value <table border="1" data-bbox="507 994 1262 1173"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Default</td> <td>Specifies Value as default value</td> </tr> <tr> <td>Fixed</td> <td>Specifies Value as fixed</td> </tr> </tbody> </table>	Option	Usage	Default	Specifies Value as default value	Fixed	Specifies Value as fixed	
Option	Usage							
Default	Specifies Value as default value							
Fixed	Specifies Value as fixed							
Nested complexType	Check this option to create an <i>XSDcomplexType</i> as a child of this global element	Complex Type 						
Nested simpleType	Check this option to create an <i>XSDsimpleType</i> as a child of this global element	Simple Type 						
Annotation	Specify the notes for this element (<i>Optional</i>)							
OK	Save the values entered and close the dialog							
Cancel	Discard the values entered and close the dialog							
Help	Display this Help topic							
UML	Open the UML Properties dialog This button is available when editing the global element	Properties 						

Notes:

- The fields **Nested complexType** and **Nested simpleType** are available in the dialog only when creating a new global element (and not when editing the global element)
- The fields **Type**, **Nested complexType** and **Nested simpleType** are mutually exclusive - selecting one disables the others
- Global elements cannot contain any UML Attributes
- Global elements cannot be the source of Association connector
- Global elements cannot be the target of a Generalization connector
- Global elements can be the target of Association connector from Complex Type or Group Class

- Global elements can have at most one Generalization connector to a Complex Type or Simple Type Class

13.1.1.3 Local Element

The **Element** icon from the **XML Schema** Toolbox creates an *XSDelement* stereotyped UML Attribute, which acts as a local XSD element, when dropped on a *XSDcomplexType* and *XSDgroup* stereotyped classes.

XSD Properties:

Double-click on the *XSDelement* stereotyped UML Attribute in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also						
Name	Specify the name for this local element							
Type	Specify either an XSD <i>built-in dataType</i> or <i>complexType</i> or <i>simpleType</i>	Complex Type <small>[1588]</small> Simple Type <small>[1591]</small>						
Reference	Specify whether to use <i>ref</i> attribute (instead of <i>type</i> attribute) to refer to the <i>complexType</i> or <i>simpleType</i> selected in the Type field in the generated XSD (<i>Optional</i>)							
Value	If Type is an XSD <i>built-in type</i> then optionally specify its value (<i>Optional</i>) <table border="1" data-bbox="502 1146 1236 1323"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Default</td> <td>Specifies Value as default value</td> </tr> <tr> <td>Fixed</td> <td>Specifies Value as fixed</td> </tr> </tbody> </table>	Option	Usage	Default	Specifies Value as default value	Fixed	Specifies Value as fixed	
Option	Usage							
Default	Specifies Value as default value							
Fixed	Specifies Value as fixed							
MinOccurs	Specify the minimum number of times this element occurs in the Class (<i>Optional</i>)							
MaxOccurs	Specify the maximum number of times this element occurs in the Class (<i>Optional</i>)							
Form	Specify whether to qualify the element (<i>Optional</i>) <table border="1" data-bbox="502 1559 1236 1765"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>qualified</td> <td>Use the Prefix supplied on the Schema package to this element</td> </tr> <tr> <td>unqualified</td> <td>Do not qualify this element</td> </tr> </tbody> </table>	Option	Usage	qualified	Use the Prefix supplied on the Schema package to this element	unqualified	Do not qualify this element	Schema Package <small>[1583]</small>
Option	Usage							
qualified	Use the Prefix supplied on the Schema package to this element							
unqualified	Do not qualify this element							
Annotation	Specify the notes for this local element (<i>Optional</i>)							
OK	Save the values entered and close the dialog							
Cancel	Discard the values entered and close the dialog							
Help	Display this Help topic							
UML	Open the UML Properties dialog	Properties <small>[696]</small>						

	This button is available when editing the local element	
--	---	--

Notes:

- Only Complex Types, Groups and Model Groups can have this UML Attribute

13.1.1.4 Global Attribute

The **Attribute** icon from the **XML Schema** Toolbox creates an *XSDtopLevelAttribute* class, which acts as an XSD global attribute, when dropped directly on a diagram.

XSD Properties:

Double-click on the *XSDtopLevelAttribute* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also						
Name	Specify the name for this global attribute							
Type	Specify either an XSD <i>built-in dataType</i> or <i>simpleType</i>							
Value	If Type is an XSD <i>built-in type</i> then optionally specify its value <table border="1" data-bbox="497 1086 1225 1261"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Default</td> <td>Specifies Value as default value</td> </tr> <tr> <td>Fixed</td> <td>Specifies Value as fixed</td> </tr> </tbody> </table>	Option	Usage	Default	Specifies Value as default value	Fixed	Specifies Value as fixed	
Option	Usage							
Default	Specifies Value as default value							
Fixed	Specifies Value as fixed							
Nested simpleType	Check this option to create an <i>XSDsimpleType</i> as a child of this global attribute	Simple Type ^[1591]						
Form	Specify whether to qualify the attribute (<i>Optional</i>) <table border="1" data-bbox="497 1413 1225 1619"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>qualified</td> <td>Use the Prefix supplied on the Schema package to this attribute</td> </tr> <tr> <td>unqualified</td> <td>Do not qualify this attribute</td> </tr> </tbody> </table>	Option	Usage	qualified	Use the Prefix supplied on the Schema package to this attribute	unqualified	Do not qualify this attribute	Schema Package ^[1583]
Option	Usage							
qualified	Use the Prefix supplied on the Schema package to this attribute							
unqualified	Do not qualify this attribute							
Annotation	Specify the notes for this attribute (<i>Optional</i>)							
OK	Save the values entered and close the dialog							
Cancel	Discard the values entered and close the dialog							
Help	Display this Help topic							
UML	Open the UML Properties dialog This button is available when editing the global attribute	Properties ^[660]						

Notes:

- The field **Nested simpleType** is available in the dialog only when creating a new global attribute (and not when editing the global attribute)
- The fields **Type** and **Nested simpleType** are mutually exclusive - selecting one disables the other
- Global attributes cannot contain any UML Attributes
- Global attributes cannot be the source of Association connector
- Global attributes cannot be the target of a Generalization connector
- Global attributes can be the target of Association connector from Complex Type Classes
- Global attributes can have at most one Generalization connector to a Simple Type Class

13.1.1.5 Local Attribute

The **Attribute** icon from the **XML Schema** Toolbox creates an *XSDattribute* stereotyped UML Attribute, which acts as a local XSD attribute, when dropped on a *XSDcomplexType* and *XSDattributeGroup* stereotyped classes.

XSD Properties:

Double-click on the *XSDattribute* stereotyped UML Attribute in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also						
Name	Specify the name for this local attribute							
Type	Specify either an XSD <i>built-in dataType</i> or <i>simpleType</i>	Simple Type ^[1591]						
Reference	Specify whether to use <i>ref</i> attribute (instead of <i>type</i> attribute) to refer to the <i>simpleType</i> selected in the Type field in the generated XSD (<i>Optional</i>)							
Value	If Type is an XSD <i>built-in type</i> then optionally specify its value (<i>Optional</i>) <table border="1" data-bbox="497 1406 1219 1581"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Default</td> <td>Specifies Value as default value</td> </tr> <tr> <td>Fixed</td> <td>Specifies Value as fixed</td> </tr> </tbody> </table>	Option	Usage	Default	Specifies Value as default value	Fixed	Specifies Value as fixed	
Option	Usage							
Default	Specifies Value as default value							
Fixed	Specifies Value as fixed							
Form	Specify whether to qualify the attribute (<i>Optional</i>) <table border="1" data-bbox="497 1653 1219 1850"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>qualified</td> <td>Use the Prefix supplied on the Schema package to this attribute</td> </tr> <tr> <td>unqualified</td> <td>Do not qualify this attribute</td> </tr> </tbody> </table>	Option	Usage	qualified	Use the Prefix supplied on the Schema package to this attribute	unqualified	Do not qualify this attribute	Schema Package ^[1583]
Option	Usage							
qualified	Use the Prefix supplied on the Schema package to this attribute							
unqualified	Do not qualify this attribute							
Annotation	Specify the notes for this local attribute (<i>Optional</i>)							
OK	Save the values entered and close the dialog							
Cancel	Discard the values entered and close the dialog							

Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the local attribute.	Properties 

Notes:

- Only Complex Types and Attribute Groups can have this UML Attribute

13.1.1.6 Attribute Group

The **Attribute Group** icon from the **XML Schema** Toolbox creates an *XSDattributeGroup* class when dropped on a diagram. This Class is used to group a set of *XSDattribute* stereotyped UML Attributes and **Simple Type** classes that can be referenced from an *XSDcomplexType* Class.

XSD Properties:

Double-click on the *XSDattributeGroup* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for this attribute group	
Annotation	Specify the notes for this attribute (Optional)	
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the attribute group	Properties 

Notes:

- Attribute Group cannot be the child of any other XSD class
- Attribute Group can contain only *XSDattribute* stereotyped UML Attributes and Simple Type classes
- Attribute Group can be the source of Association connector to another Attribute Group
- Attribute Group can be the target of Association connector from Complex Type classes
- Attribute Group cannot be the source or target of a Generalization connector

13.1.1.7 Complex Type

The **Complex Type** icon from the **XML Schema** Toolbox creates an *XSDcomplexType* stereotyped Class when dropped on a diagram.

XSD Properties:

Double-click on the *XSDcomplexType* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also								
Name	Specify the name for this complexType									
Model Group	Specify how the child elements of this complexType should occur in the Schema <table border="1" data-bbox="507 539 1276 801"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>sequence</td> <td>Specifies that the child elements must occur in the specified order</td> </tr> <tr> <td>choice</td> <td>Specifies that only one of the child elements must occur</td> </tr> <tr> <td>all</td> <td>Specifies that the child elements can occur in any order</td> </tr> </tbody> </table>	Option	Usage	sequence	Specifies that the child elements must occur in the specified order	choice	Specifies that only one of the child elements must occur	all	Specifies that the child elements can occur in any order	
Option	Usage									
sequence	Specifies that the child elements must occur in the specified order									
choice	Specifies that only one of the child elements must occur									
all	Specifies that the child elements can occur in any order									
MinOccurs	Specify the minimum number of times this element occurs in the Class (<i>Optional</i>)									
MaxOccurs	Specify the maximum number of times this element occurs in the Class (<i>Optional</i>)									
Annotation	Specify the notes for this element (<i>Optional</i>)									
Abstract	Specify whether this complexType can be used in the instance XML file (<i>Optional</i>) Default: False									
Mixed	Specify whether character data can appear among the child elements (<i>Optional</i>) Default: False									
OK	Save the values entered and close the dialog									
Cancel	Discard the values entered and close the dialog									
Help	Display this Help topic									
UML	Open the UML Properties dialog This button is available when editing the complexType	Properties 660								

Notes:

- Complex Types can contain both *XSDelement* and *XSDattribute* stereotyped UML Attributes
- Complex Types can contain other Complex Types as child elements
- Complex Types can be children of Global Elements
- Complex Types can have Association connectors to other Complex Type, Simple Type, Attribute Group, Group and Model Group
- Complex Types can have at most one Generalization connector to either another Complex Type or a Simple Type Class

13.1.1.8 Model Group

The **Model Group** icon from the **XML Schema** Toolbox creates an *XSDsequence*, *XSDchoice* or *XSDall* stereotyped Class when dropped on a diagram.

XSD Properties:

Double-click on the **Model Group** in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also								
Name	Specify the name for this Model Group									
Model Group	Specify how the child elements should occur. <table border="1" data-bbox="497 813 1284 1211"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>sequence</td> <td>This option creates an <i>XSDsequence</i> stereotyped Class. It specifies that the child elements must occur in the specified order.</td> </tr> <tr> <td>choice</td> <td>This option creates an <i>XSDchoice</i> stereotyped Class. It specifies that only one of the child elements must occur.</td> </tr> <tr> <td>all</td> <td>This option creates an <i>XSDall</i> stereotyped Class. It specifies that the child elements can occur in any order.</td> </tr> </tbody> </table>	Option	Usage	sequence	This option creates an <i>XSDsequence</i> stereotyped Class. It specifies that the child elements must occur in the specified order.	choice	This option creates an <i>XSDchoice</i> stereotyped Class. It specifies that only one of the child elements must occur.	all	This option creates an <i>XSDall</i> stereotyped Class. It specifies that the child elements can occur in any order.	
Option	Usage									
sequence	This option creates an <i>XSDsequence</i> stereotyped Class. It specifies that the child elements must occur in the specified order.									
choice	This option creates an <i>XSDchoice</i> stereotyped Class. It specifies that only one of the child elements must occur.									
all	This option creates an <i>XSDall</i> stereotyped Class. It specifies that the child elements can occur in any order.									
MinOccurs	Specify the minimum number of times this Model Group occurs in the Class (<i>Optional</i>)									
MaxOccurs	Specify the maximum number of times this Model Group occurs in the Class (<i>Optional</i>)									
Annotation	Specify the notes for this Model Group (<i>Optional</i>)									
OK	Save the values entered and close the dialog									
Cancel	Discard the values entered and close the dialog									
Help	Display this Help topic									
UML	Open the UML Properties dialog This button is available when editing the Model Group	Properties (660)								

Notes:

- Model Groups can contain only XSDelement stereotyped UML Attributes
- Model Groups can contain other Complex Types and Simple Types as child elements
- Model Groups can have Association connector to other Complex Type, Simple Type, Group and Model Group
- Model Groups must have at least one incoming Association connector from a Complex Type
- Model Groups cannot be the source or target of Generalization connectors

13.1.1.9 Group

The **Group** icon from the **XML Schema** Toolbox creates an *XSDgroup* stereotyped Class when dropped on a diagram. This Class is used to group a set of *XSDelement* stereotyped UML Attributes, **Complex Type** and **Simple Type** classes that can be referenced from an *XSDcomplexType* Class.

XSD Properties:

Double-click on the *XSDgroup* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also								
Name	Specify the name for this group									
Model Group	Specify how the child elements of this group should occur in the Complex Type	Complex Type [1588]								
	<table border="1"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>sequence</td> <td>Specifies that the child elements must occur in the specified order</td> </tr> <tr> <td>choice</td> <td>Specifies that only one of the child elements must occur</td> </tr> <tr> <td>all</td> <td>Specifies that the child elements can occur in any order</td> </tr> </tbody> </table>		Option	Usage	sequence	Specifies that the child elements must occur in the specified order	choice	Specifies that only one of the child elements must occur	all	Specifies that the child elements can occur in any order
	Option		Usage							
	sequence		Specifies that the child elements must occur in the specified order							
choice	Specifies that only one of the child elements must occur									
all	Specifies that the child elements can occur in any order									
Annotation	Specify the notes for this group (<i>Optional</i>)									
OK	Save the values entered and close the dialog									
Cancel	Discard the values entered and close the dialog									
Help	Display this Help topic									
UML	Open the UML Properties dialog This button is available when editing the group.	Properties [660]								

Notes:

- Group can contain only *XSDelement* stereotyped UML Attributes
- Group can contain Complex Types and Simple Types as child elements
- Group can have Association connector to other Complex Types, Simple Types and Groups
- Group can be the target of Association connector from Complex Types
- Group cannot be the source or target of a Generalization connector

13.1.1.10 Simple Type

The **Simple Type** icon from the **XML Schema** Toolbox creates an *XSDsimpleType* class when dropped on a diagram.

XSD Properties:

Double-click on the *XSDsimpleType* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also						
Name	Specify the name for this simpleType							
Type	Specify either an XSD <i>built-in dataType</i> or simpleType <table border="1" data-bbox="491 510 1276 757"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Restriction</td> <td>Restricts the value of this simpleType to the selected Type The various restrictions (facets) on the simpleType are a tagged-values on this Class.</td> </tr> <tr> <td>List</td> <td>Specifies this simpleType as a list of values of the selected</td> </tr> </tbody> </table>	Field	Usage	Restriction	Restricts the value of this simpleType to the selected Type The various restrictions (facets) on the simpleType are a tagged-values on this Class.	List	Specifies this simpleType as a list of values of the selected	
Field	Usage							
Restriction	Restricts the value of this simpleType to the selected Type The various restrictions (facets) on the simpleType are a tagged-values on this Class.							
List	Specifies this simpleType as a list of values of the selected							
Annotation	Specify the notes for this simpleType (<i>Optional</i>)							
OK	Save the values entered and close the dialog							
Cancel	Discard the values entered and close the dialog							
Help	Display this Help topic							
UML	Open the UML Properties dialog This button is available when editing the simpleType	Properties [660]						

Notes:

- Simple Types cannot contain any XSDelement or XSDattribute stereotyped UML Attributes
- Simple Types cannot contain any child classes
- Simple Types cannot be the source of an Association connector
- Simple Types can be the target of a Generalization connector
- Simple Types can have at most one Generalization connector to another Simple Type Class

13.1.1.11 Union

The **Union** icon from the **XML Schema** Toolbox creates an *XSDunion* class when dropped directly on a diagram. This Class, which is a Simple Type, defines a collection of Simple Types.

XSD Properties:

Double-click on the *XSDunion* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for this Union	
Member Types	Specify the collection of XSD <i>built-in dataTypes</i> and simpleTypes	Simple Type [1591]
Annotation	Specify the notes for this Union (<i>Optional</i>)	

OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the Union	Properties 660

Notes:

- Unions cannot contain any child Classes
- Unions cannot contain any XSDelement or XSDattribute stereotyped UML Attributes
- Unions cannot be the source of an Association connector
- Unions can be the target of Association connector from Complex Types
- Unions can be the target of a Generalization connector from Simple Types

13.1.1.12 Enumeration

The **Enum** icon from the **XML Schema** Toolbox creates an *enumeration* stereotyped Class when dropped on a diagram. This Class, which is a Simple Type, defines a list of acceptable values for this Class.

XSD Properties:

Double-click on the *enumeration* stereotyped Class under a Schema package in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for this Enumeration	
Type	Specify either an XSD <i>built-in dataType</i> or simpleType	Simple Type 1591
Values	Specify the list of acceptable values for the specified Type	
Annotation	Specify the notes for this Enumeration (<i>Optional</i>)	
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the Enumeration	Properties 660

Notes:

- Enumeration cannot contain any XSDelement or XSDattribute stereotyped UML Attributes
- Enumeration cannot contain any child Classes
- Enumeration cannot be the source of an Association connector
- Enumeration can be the target of a Generalization connector

- Enumeration can have at most one Generalization connector to another Simple Type Class

13.1.1.13 Any

The **Any** icon from the **XML Schema** Toolbox creates an *XSDany* stereotyped Class when dropped on a diagram. This Class allows a Complex Type to contain elements that are not specified in the Schema package.

XSD Properties:

Double-click on the *XSDany* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also								
Name	Specify the name for this Class									
Namespace	Specifies the namespace that contains the elements that can be used in the Complex Type (<i>Optional</i>)	Complex Type [1588]								
ProcessContents	Specifies how the XML Parser should validate these elements (<i>Optional</i>) <table border="1" data-bbox="497 1003 1225 1256"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>lax</td> <td>Tries to validate the elements against their Schema is flagged when the Schema cannot be obtained</td> </tr> <tr> <td>skip</td> <td>Skips validating the elements</td> </tr> <tr> <td>strict</td> <td>Validates the elements against their Schema</td> </tr> </tbody> </table>	Option	Usage	lax	Tries to validate the elements against their Schema is flagged when the Schema cannot be obtained	skip	Skips validating the elements	strict	Validates the elements against their Schema	
Option	Usage									
lax	Tries to validate the elements against their Schema is flagged when the Schema cannot be obtained									
skip	Skips validating the elements									
strict	Validates the elements against their Schema									
Annotation	Specify the notes for this Class (<i>Optional</i>)									
OK	Save the values entered and close the dialog									
Cancel	Discard the values entered and close the dialog									
Help	Display this Help topic									
UML	Open the UML Properties dialog This button is available when editing this Class	Properties [660]								

Notes:

- Any Class cannot contain any UML Attributes or child XSD Classes
- Any Class cannot be the child of any XSD Class
- Any Class cannot be the target of Generalization connectors
- Any Class cannot be the source of Association and Generalization connectors
- Any Class can be the target of Association connectors from Complex Types, Groups and Model Groups
- Any Class must have at least one incoming Association connector

13.1.1.14 Any Attribute

The **Any Attribute** icon from the **XML Schema** Toolbox creates an *XSDany* stereotyped UML Attribute when dropped on Complex Type or an Attribute Group. This Class allows a Complex Type or an Attribute Group to contain elements that are not specified in the Schema package.

XSD Properties:

Double-click on the *XSDany* stereotyped UML Attribute in a Class or the **Project Browser**

Reference:

Field/Button	Usage	See Also								
Name	Specify the name for this UML Attribute									
Namespace	Specifies the namespace that contains the attributes that can be used in the Complex Type or Attribute Group (<i>Optional</i>)	Complex Type <small>[1588]</small> Attribute Group <small>[1588]</small>								
ProcessContents	Specifies how the XML Parser should validate these attributes (<i>Optional</i>) <table border="1" data-bbox="496 1016 1227 1279"> <thead> <tr> <th>Option</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>lax</td> <td>Tries to validate the attribute against their Schema; if it is flagged when the Schema cannot be obtained</td> </tr> <tr> <td>skip</td> <td>Skips validating the attributes</td> </tr> <tr> <td>strict</td> <td>Validates the attributes against their Schema</td> </tr> </tbody> </table>	Option	Usage	lax	Tries to validate the attribute against their Schema; if it is flagged when the Schema cannot be obtained	skip	Skips validating the attributes	strict	Validates the attributes against their Schema	
Option	Usage									
lax	Tries to validate the attribute against their Schema; if it is flagged when the Schema cannot be obtained									
skip	Skips validating the attributes									
strict	Validates the attributes against their Schema									
Annotation	Specify the notes for this UML Attribute (<i>Optional</i>)									
OK	Save the values entered and close the dialog									
Cancel	Discard the values entered and close the dialog									
Help	Display this Help topic									
UML	Open the UML Properties dialog This button is available when editing this UML Attribute	Properties <small>[696]</small>								

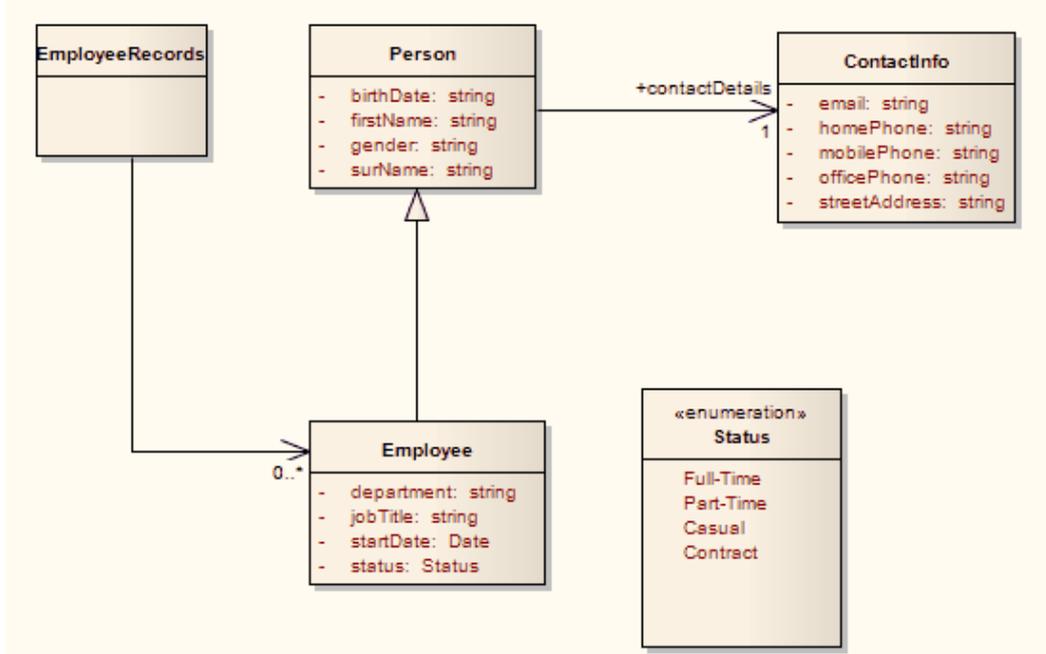
Notes:

- Only Complex Types and Attribute Groups can have this UML Attribute

13.1.2 Abstract XSD models

XML schemas can also be modeled using simple, abstract Class models. This can be useful in enabling an architect to start working at a higher level of abstraction, without concern for the implementation details of a Schema. Such an abstract model can be refined further using the **XML Schema** pages of the **Toolbox**, or it can be generated directly by Enterprise Architect's **Schema Generator**. In this case, a set of **default mappings** is assumed by the schema generator to convert the abstract model to an XSD file.

Example:

Topic	Detail
Diagram	<p>The following is a simplified version of the <i>Employee Details</i> example model, which does not use XSD-specific stereotypes or Tagged Values:</p>  <pre> classDiagram class EmployeeRecords class Person { - birthDate: string - firstName: string - gender: string - surName: string } class Employee { - department: string - jobTitle: string - startDate: Date - status: Status } class ContactInfo { - email: string - homePhone: string - mobilePhone: string - officePhone: string - streetAddress: string } class Status { <<enumeration>> Full-Time Part-Time Casual Contract } EmployeeRecords --> Employee : 0..* Employee < -- Person Person --> ContactInfo : +contactDetails, 1 </pre>
Schema	<p>The following schema fragment would be generated by Enterprise Architect, given the above model:</p> <pre> <?xml version="1.0"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:simpleType name="Status"> <xs:restriction base="xs:string"> <xs:enumeration value="Full-Time"/> <xs:enumeration value="Part-Time"/> <xs:enumeration value="Casual"/> <xs:enumeration value="Contract"/> </xs:restriction> </xs:simpleType> <xs:element name="Person" type="Person"/> <xs:complexType name="Person"> <xs:sequence> <xs:element name="firstName" type="xs:string"/> <xs:element name="surName" type="xs:string"/> <xs:element name="birthDate" type="xs:string"/> <xs:element name="gender" type="xs:string"/> <xs:element name="contactDetails" type="ContactInfo"/> </xs:sequence> </xs:complexType> <xs:element name="Employee" type="Employee"/> <xs:complexType name="Employee"> <xs:complexContent> <xs:extension base="Person"> <xs:sequence> <xs:element name="status" type="Status"/> <xs:element name="jobTitle" </pre>

Topic	Detail
	<pre> type="xs:string" /> type="xs:date" /> type="xs:string" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> <xs:element name="EmployeeRecords" type="EmployeeRecords" /> <xs:complexType name="EmployeeRecords"> <xs:sequence> <xs:element name="Employee" type="Employee" minOccurs="0" maxOccurs="unbounded" /> </xs:sequence> </xs:complexType> <xs:element name="ContactInfo" type="ContactInfo" /> <xs:complexType name="ContactInfo"> <xs:sequence> <xs:element name="homePhone" type="xs:string" /> <xs:element name="mobilePhone" type="xs:string" /> > <xs:element name="officePhone" type="xs:string" /> > <xs:element name="email" type="xs:string" /> <xs:element name="streetAddress" type="xs: string" /> </xs:sequence> </xs:complexType> </xs:schema> </pre>

Learn More:

- [Default UML to XML Mappings](#) ^[1597]
- [Generate XSD](#) ^[1598]

13.1.2.1 Default UML to XSD Mappings

The following table describes the default mapping of UML to XSD constructs. This set of mappings is useful when defining simple schemas from abstract Class models. The defaults are also assumed by the schema generator when generating unsteretyped elements in an abstract model. The **XML Schema** pages of the **Toolbox** (and UML Profile for XSD) override these default mappings through the use of stereotypes and Tagged Values.

UML Construct	Default XSD Production Rules
Package	<p>A schema element is generated for the target package. If the target package includes Classes from another package, which has the Tagged Values <i>targetNamespace</i> and <i>targetNamespacePrefix</i> set, these are included as attributes of the schema element.</p> <p>In addition, an <i>import</i> or <i>include</i> element is created for each referenced package. (An <i>include</i> element is used if the external package shares the same <i>targetNamespace</i> Tagged Value as the target package. An <i>import</i> element is used where the <i>targetNamespaces</i> differ).</p>
Class	<p>A root-level element declaration and <i>complexType</i> definition are generated. The element name and type are the same as the Class name. An XSD sequence model group is generated to contain UML attributes generated as elements.</p>
Attribute	<p>An element is declared for each Class attribute. The element name is set to that of the UML attribute name. This is prefixed with the Class name to make the element</p>

UML Construct	Default XSD Production Rules
	unique. The <i>minOccurs</i> and <i>maxOccurs</i> attributes are set to reflect the attribute cardinality. If the attribute refers to another Class, the element declaration is followed a <i>complexType</i> definition, which contains a reference to the appropriate <i>complexType</i> .
Association	An element is declared for each association owned by a Class. The element name is set to that of the association role. The <i>minOccurs</i> and <i>maxOccurs</i> reflect the cardinality of the association.
Generalization (Inheritance)	For single inheritances, an extension element is generated with the base attribute set to the base Classname. The UML attributes of the child Class are then appended to an all model group within the extension element.
«enumeration» (stereotype)	A <i>simpleType</i> element is declared for the enumeration Class with the name attribute set to the Classname. A restriction element is generated with base set to string. Each of the Class attributes is appended to the restriction element as XSD enumeration elements with value set to the UML attribute name. Any type specification for the UML attributes is ignored by the schema generator.

Notes:

- If left unspecified, *minOccurs* and *maxOccurs* default to 1
- If the direction of the association is unspecified, the owner is assumed to be the source

13.1.3 Generate XSD

The **Generate XML Schema** feature forward-engineers a UML Class model to a W3C XML Schema (XSD) file. An XML schema corresponds to a UML package in Enterprise Architect, therefore XML schema generation is a package-level operation.

Access:

- **Tools | XML Schema | Generate XML Schema**
- Right-click on a package in the **Project Browser** and select the following context menu : **Code Engineering | Generate XML Schema**

Reference:

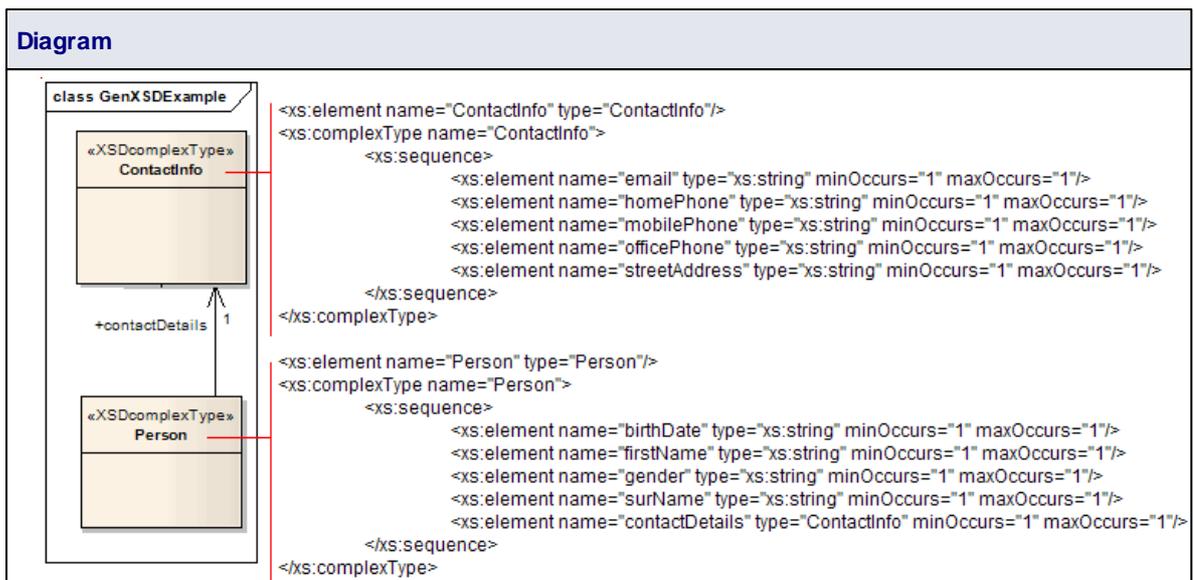
Field/Option/ Button	Usage	See Also				
Encoding	Specify the required XML encoding					
XSD Style	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Generate global element for all global ComplexTypes ('Garden of Eden' style)</td> <td>Generate Schema in the Gal This option is checked, by de</td> </tr> </tbody> </table>	Field	Usage	Generate global element for all global ComplexTypes ('Garden of Eden' style)	Generate Schema in the Gal This option is checked, by de	Garden of Eden Style <small>[1599]</small>
	Field	Usage				
Generate global element for all global ComplexTypes ('Garden of Eden' style)	Generate Schema in the Gal This option is checked, by de					
<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Field	Usage				
Field	Usage					
Referenced Package Options	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Field	Usage			
	Field	Usage				
<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Field	Usage				
Field	Usage					

Field/Option/ Button	Usage	See Also						
	<table border="1"> <tr> <td>Generate XSD for Referenced packages</td> <td>Generate Schema for packages referenced by any of the packages in the list box</td> </tr> <tr> <td>Prompt when missing Filename</td> <td>Enable Enterprise Architect to prompt for a filename for a referenced package during Schema generation, if the package's Schema file is missing</td> </tr> <tr> <td>Use relative path to reference XSDs (if 'schemaLocation' tag is empty)</td> <td>Use a relative-path in the XSD <i>include</i> statement when referencing packages, provided that the path is empty on the referenced package. Ensure that the Schema File XSD Schema Properties dialog Properties dialog for a Schema referenced and referencing XSDs is correctly determined.</td> </tr> </table>	Generate XSD for Referenced packages	Generate Schema for packages referenced by any of the packages in the list box	Prompt when missing Filename	Enable Enterprise Architect to prompt for a filename for a referenced package during Schema generation, if the package's Schema file is missing	Use relative path to reference XSDs (if 'schemaLocation' tag is empty)	Use a relative-path in the XSD <i>include</i> statement when referencing packages, provided that the path is empty on the referenced package. Ensure that the Schema File XSD Schema Properties dialog Properties dialog for a Schema referenced and referencing XSDs is correctly determined.	
Generate XSD for Referenced packages	Generate Schema for packages referenced by any of the packages in the list box							
Prompt when missing Filename	Enable Enterprise Architect to prompt for a filename for a referenced package during Schema generation, if the package's Schema file is missing							
Use relative path to reference XSDs (if 'schemaLocation' tag is empty)	Use a relative-path in the XSD <i>include</i> statement when referencing packages, provided that the path is empty on the referenced package. Ensure that the Schema File XSD Schema Properties dialog Properties dialog for a Schema referenced and referencing XSDs is correctly determined.							
Child Package Options	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Generate XSD for Child Packages</td> <td> Generate schema for child packages selected package. <ul style="list-style-type: none"> • Include all packages - includes all packages under the package in the list box • Include <XSDschem - includes only those packages with the stereotype «XSDschem» </td> </tr> </tbody> </table> <p>The list-box lists, for each package, the package name and the file path of the file to be generated.</p> <p>Change the file path for a package by double-click on the entry in the list box.</p> <p>Ensure that the checkbox is selected for each package required for generation.</p>	Field	Usage	Generate XSD for Child Packages	Generate schema for child packages selected package. <ul style="list-style-type: none"> • Include all packages - includes all packages under the package in the list box • Include <XSDschem - includes only those packages with the stereotype «XSDschem» 			
Field	Usage							
Generate XSD for Child Packages	Generate schema for child packages selected package. <ul style="list-style-type: none"> • Include all packages - includes all packages under the package in the list box • Include <XSDschem - includes only those packages with the stereotype «XSDschem» 							
Generate	Generate the Schema for each of the packages checked in the list-box.							
Close	Close this dialog							
Help	Display this Help topic							
View Schema	View the generated Schema for a package highlighted in the list-box.							
Progress	Lists the progress of Schema Generation.							

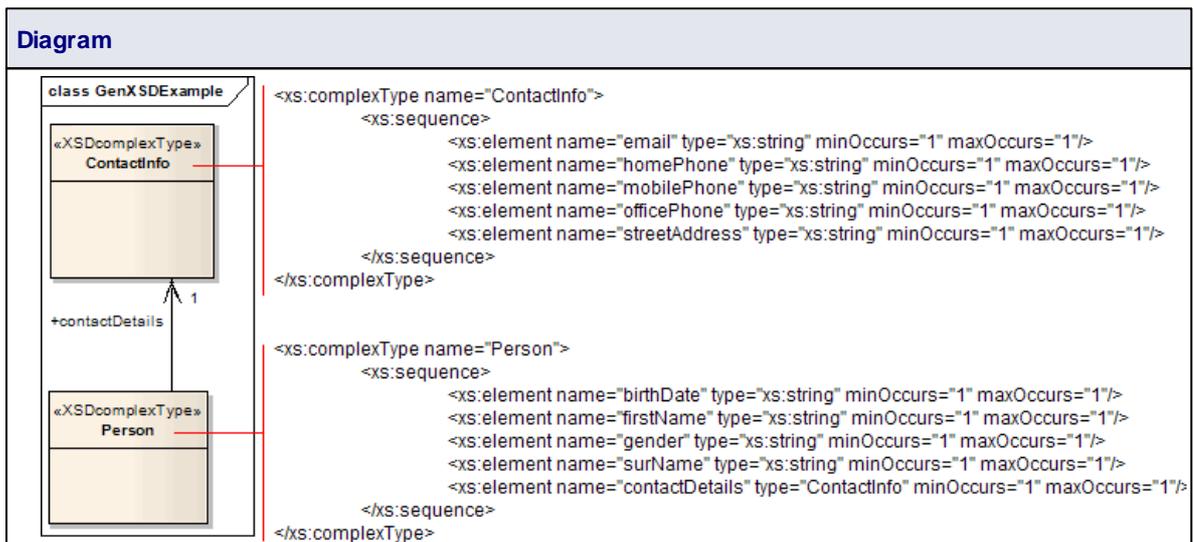
13.1.3.1 Generate Global Element

Enterprise Architect, by default, generates XML Schema in the **Garden of Eden** style. For every global *XSDcomplexType* stereotyped Class, Enterprise Architect generates a global element.

Example:



You can change the above specified default behaviour by de-selecting the **Generate global element for all global ComplexTypes** checkbox on the **Generate XML Schema** dialog. Then, the generated XSD no longer contains the global element.



Learn More:

- [Generate XML Schema](#)¹⁵⁹⁸

13.1.4 Import XSD

The **XML Schema Import** facility is used to reverse engineer a W3C XML Schema (XSD) file as a UML Class model. An XSD file is imported into Enterprise Architect as a UML package.

Access:

- **Tools | XML Schema | Import XML Schema**
- Right-click on a package in the **Project Browser** and select the following context menu : **Code Engineering | Import XML Schema**

Reference:

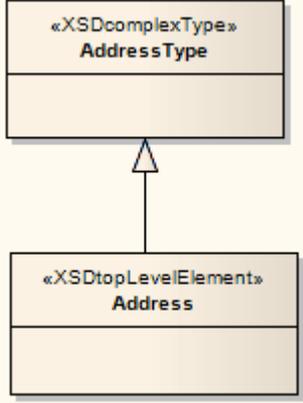
Field/Option/Button	Usage	See also										
Root Package	Indicates the name of the package selected.											
Directory	Specifies the directory where the XSD file(s) is located.											
Selected File(s)	Lists the XML Schema(s) currently selected for import. To select several individual files CTRL + left click on each file. To select a range of files, press SHIFT and select the first and last file in the range.											
Import Options	<table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Import global elements with "Type" postfix</td> <td>Treat the global elements as <i>ComplexType</i> if it is referenced as separate entities.</td> </tr> <tr> <td>Import referenced XML Schema(s)</td> <td>Import any XML Schema referenced by any of the Selected File(s) field.</td> </tr> <tr> <td>Create Diagram for XML Schema(s)</td> <td>Creates a Class diagram for the imported <i>XSDschema</i>.</td> </tr> <tr> <td>Import XSD Elements/Attributes as</td> <td>Indicate how the inline <i>XSDattributes</i> are imported.</td> </tr> </tbody> </table>	Field	Usage	Import global elements with "Type" postfix	Treat the global elements as <i>ComplexType</i> if it is referenced as separate entities.	Import referenced XML Schema(s)	Import any XML Schema referenced by any of the Selected File(s) field.	Create Diagram for XML Schema(s)	Creates a Class diagram for the imported <i>XSDschema</i> .	Import XSD Elements/Attributes as	Indicate how the inline <i>XSDattributes</i> are imported.	Global Element and ComplexType
Field	Usage											
Import global elements with "Type" postfix	Treat the global elements as <i>ComplexType</i> if it is referenced as separate entities.											
Import referenced XML Schema(s)	Import any XML Schema referenced by any of the Selected File(s) field.											
Create Diagram for XML Schema(s)	Creates a Class diagram for the imported <i>XSDschema</i> .											
Import XSD Elements/Attributes as	Indicate how the inline <i>XSDattributes</i> are imported.											
Import	Initiate XSD Import.											
Close	Close this dialog.											
Help	Display this Help topic											
Progress	Lists the progress of Schema Import.											

Notes:

- If an XML Schema file being imported already exists in the model, Enterprise Architect skips importing the file
- References to XSD *Primitive Types* are always imported as UML Attributes
- References to XSD constructs in external Schema files are always imported as UML Attributes
- Enterprise Architect uses the *schemaLocation* attribute in the *XSD Import* and *XSD Include* elements of an XML Schema to determine the dependencies between the files. Ensure that this attribute is set to a valid file path (and not a URL) for the dependent XML Schema(s) to be imported correctly

13.1.4.1 Global Element and ComplexType

Topic	Detail
ComplexType element naming convention	Some XML Schemas have ComplexType elements with the same name as the referring global elements, but with the suffix <i>Type</i> as shown below: <pre><xs:element name="Address" type="AddressType" /> <xs:complexType name="AddressType" /></pre>

Topic	Detail
	<pre data-bbox="531 309 866 365"><xs: sequence / > </ xs: compl exType ></pre> <p data-bbox="483 369 1409 454">On XSD import, Enterprise Architect treats this global element and its bounding ComplexType as a single entity and creates a single XSDcomplexType stereotyped Class with the same name as the global element as shown below:</p> 
<p data-bbox="260 689 451 745">Changing the default behavior</p>	<p data-bbox="483 689 1409 902">You can change the above specified default behaviour by selecting the Import global elements with "Type" postfix checkbox. When you select this option, Enterprise Architect treats the global element and the ComplexType it is referring to as two separate entities. So, for the above example, Enterprise Architect creates an <i>XSDtopLevelElement</i> stereotyped Class for the global element and an <i>XSDcomplexType</i> stereotyped Class for the ComplexType, and connects them as follows:</p> 

Notes:

- Enterprise Architect treats the following as two separate entities irrespective of whether the **Import global elements with "Type" postfix** checkbox is selected or unselected:

```
<xs: element name=" HomeAddress" type=" AddressType" / >
<xs: complexType name=" AddressType" >
  <xs: sequence / >
</ xs: compl exType >
```

Learn More:

- [Import XML Schema](#)^[1600]

13.1.5 UML Profile for XSD

The **UML Profile for XSD** specifies a set of stereotypes, tagged values and constraints that can be applied to the UML model in order to change particular aspects of the resulting schema. For example, you might have to convert certain UML Class attributes to XSD attributes, or use a model group other than the default *Sequence*.

Enterprise Architect provides native support for the UML Profile for XSD via the **XML schema** page of the Diagram Toolbox. Alternatively, you can use the profile via Enterprise Architect's generic profile mechanism by downloading the UML Profile for XSD. The XSD profile used by Enterprise Architect is an adaptation of the profile defined in *Modeling XML Applications with UML* (David Carlson).

Learn More:

- [Profile for XML Schema](#)^[1603]
- [XSD Datatypes Package](#)^[1610]

13.1.5.1 Profile for XML Schema

The **UML Profile for XSD** specifies stereotypes that provide an explicit mapping from XSD to UML constructs. The Tagged Values further define aspects of the mapping, such as whether the elements should be qualified. The constraints define any conditions that must be satisfied for the stereotype to apply.

The UML Profile for XSD specifies the following stereotypes:

- [XSDschema](#)^[1603]
- [XSDcomplexType](#)^[1604]
- [XSDsimpleType](#)^[1605]
- [XSDsequence](#)^[1605]
- [XSDchoice](#)^[1606]
- [XSDelement](#)^[1606]
- [XSDattribute](#)^[1607]
- [XSDany](#)^[1607]
- [XSDrestriction](#)^[1608]
- [XSDgroup](#)^[1608]
- [XSDtopLevelElement](#)^[1608]
- [XSDtopLevelAttribute](#)^[1609]
- [XSDunion](#)^[1609]
- [XSDattributeGroup](#)^[1609]

The following tables list the features of the UML Profile for XSD.

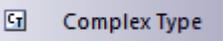
- Tagged Value names are shown in bold followed by the allowed values.
- If a default value is used by Enterprise Architect's schema generator, it is underlined.

«XSDschema»:

UML Construct		Package
Description		All Classes in a package are defined within one schema. This stereotype can be used to specify schema-wide settings.
Toolbox Icon		 Schema
Tagged Values	anonymousRole (true <u>false</u>)	Specifies if the role name is included in the element declaration for the UML attribute.
	anonymousType (true <u>false</u>)	Specifies whether the Class type is anonymous for attributes.

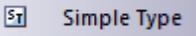
	attributeFormDefault (qualified <u>unqualified</u>)	Determines whether attribute instances must be qualified.
	defaultNamespace	The default namespace used in this schema. This value is used to specify the default namespace attribute (<i>xmlns=</i>), in the schema element.
	elementDerivation (true false)	Determines whether inheritances are generated using XSD extension or copy-down inheritance.
	elementFormDefault (qualified <u>unqualified</u>)	Determines whether element instances must be qualified.
	memberNames (qualified unqualified)	Determines whether elements generated from Class attributes have their name qualified by the corresponding Class name.
	modelGroup (all <u>sequence</u> choice)	Specifies the default XSD model group used to generate <i>complexType</i> definitions.
	schemaLocation	The URI that identifies the location of the schema. This value is used in the import and include elements.
	targetNamespace	The URI that uniquely identifies this schema's namespace.
	targetNamespacePrefix	The prefix that abbreviates the <i>targetNamespace</i> .
	version	The version of this schema.
	xmlns	The comma separated list of all the namespaces (and their prefixes).
Constraints		None.

«XSDcomplexType»:

UML Construct		Class
Description		<i>complexType</i> definitions are created for generic UML Classes. This stereotypes helps tailor the generation of a <i>complexType</i> definition.
Toolbox Icon		 Complex Type
Tagged Values	memberNames (qualified unqualified)	Determines whether elements generated from the UML Class attributes and associations have their name qualified by the corresponding Class name for this <i>complexType</i> definition.
	mixed (true <u>false</u>)	Determines whether this element can contain mixed element and character content. See the W3CXML Schema recommendation.
	modelGroup (all sequence choice)	Overrides the default XSD model for generating this <i>complexType</i> definition.
	minOccurs	Specifies the minimum number of times the model group in the element can occur.
	maxOccurs	Specifies the maximum number of times the model group in the element can occur.

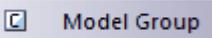
	nillable (true false)	Specifies whether a null value can be assigned to the top-level element generated for this <i>complexType</i> element.
Constraints		None.

«XSDsimpleType»

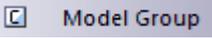
UML Construct		Class
Description		An XSD <i>simpleType</i> is generated for Classes with this stereotype.
Toolbox Icon		 Simple Type
Tagged Values	derivation: (restriction list)	Specifies the derivation of the <i>simpleType</i> . See the W3C XML Schema recommendation.
	length:	See the W3C XML Schema recommendation.
	minLength:	
	maxLength:	
	minInclusive:	
	minExclusive:	
	maxInclusive:	
	maxExclusive:	
	totalDigits:	
	fractionDigits:	
	whiteSpace:	
	pattern:	
Constraints		This Class can only participate in an inheritance relation with another <i>simpleType</i> . It cannot have any attributes or own any associations; they are ignored if present.

«XSDsequence»

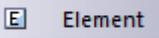
UML Construct		Class
Description		The schema generator creates a sequence model group as the container for the attributes and associations owned by this Class. The model group is in turn added to the model groups of this Class respective owners. Tagged values specified by owners of this Class persist through to the child elements of this model group. Thus if

		<i>memberNames</i> are unqualified for a <i>complexType</i> , so are the children of this model group when added to that <i>complexType</i> .
Toolbox Icon		
Tagged Values		None.
Constraints		This Class must be the destination of unidirectional associations. If it is not, this Class and its connectors are ignored, possibly invalidating other model group Classes. Inheritance relations are ignored for this Class.

«XSDchoice»

UML Construct		Class
Description		Creates an XSD choice element. See <i>XSDsequence</i> for more details.
Toolbox Icon		
Tagged Values		None.
Constraints		As for <i>XSDsequence</i> .

«XSDElement»

UML Construct		UML Attribute, Association <i>End</i>
Description		By applying this stereotype to a UML Class or Association <i>End</i> , the corresponding UML entity is generated as an element within the parent <i>complexType</i> and not as an XSD attribute.
Toolbox Icon		
Tagged Values	form: (qualified unqualified)	Overrides the schema's <i>elementFormDefault</i> value.
	position:	Causes the elements to be ordered within a sequence model group of the containing <i>complexType</i> . Duplicated and invalid position Tagged Values are ignored and result in undefined ordering of the UML attributes. Missing position values cause the defined positions to be allocated as specified, with the remaining elements filling the missing positions in an undefined order.
	anonymousRole: (true false)	Specifies if the role name is included in the element declaration for the UML attribute.
	anonymousType: (true false)	Specifies whether the Class type is anonymous for attributes.

	nillable (true false)	Specifies whether a null value can be assigned to this element.
	default	See the W3C XML Schema recommendation.
	fixed	
Constraints		None.

«XSDattribute»

UML Construct		UML Attribute, Association <i>End</i>
Description		By applying this stereotype to a UML Class attribute or Association <i>End</i> , the corresponding UML entity is generated as an XSD attribute within the parent <i>complexType</i> and not as an XSD element.
Toolbox Icon		 Attribute
Tagged Values	form: (qualified unqualified)	Overrides the schema's <i>attributeFormDefault</i> value.
	use: (prohibited optional required)	See the W3C XML Schema recommendation.
	default	
	fixed	
Constraints		The attribute <i>datatype</i> should not see a Class specification, otherwise it is ignored.

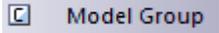
«XSDany»

UML Construct		UML Attribute
Description		If applied to a UML attribute, an XSD <i>anyAttribute</i> element is generated. If applied to a UML Class, an XSD <i>any</i> element is generated.
Toolbox Icon		 Any Attribute
Tagged Values	namespace:	See the W3C XML Schema recommendation.
	processContents: (skip lax strict)	
Constraints		None.

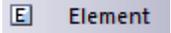
«XSDrestriction»

UML Construct		Generalization
Description		Overrides the default use of XSD extension for inheritance and generates the child as a <i>complexType</i> with a restriction element instead.
Tagged Values		None.
Constraints		Applies only to UML Class parent-child relations.

«XSDgroup»

UML Construct		Class
Description		An <i>XSDgroup</i> is generated for Classes with this stereotype.
Toolbox Icon		
Tagged Values	modelGroup: (<u>sequence</u> choice all)	Overrides the default XSD model for generating this group definition.
Constraints		A group Class can only associate itself to other group Classes. A group Class can be associated by another group Class or a <i>complexType</i> Class. The association should be via an Association connector. A group Class cannot be inherited/aggregated.

«XSDtopLevelElement»

UML Construct		Class
Description		Creates an <code><xs:element></code> construct which acts as a container for <i>XSDcomplexType</i> and <i>XSDsimpleType</i> Class.
Toolbox Icon		
Tagged Values	default	See the W3C XML Schema recommendation.
	fixed	
Constraints		An <i>XSDtopLevelElement</i> Class can contain either an <i>XSDsimpleType</i> or an <i>XSDcomplexType</i> as its child Class. When such a Class is present as its child, all its inheritance is ignored. This Class cannot be inherited.

«XSDtopLevelAttribute»

UML Construct		Class
Description		Creates an <code><xs:attribute></code> construct which acts as a container for <i>XSDsimpleType</i> Class.
Toolbox Icon		 Attribute
Tagged Values	use: (<i>optional</i> required prohibited)	See the W3C XML Schema recommendation.
	default	
	fixed	
Constraints		An <i>XSDtopLevelAttribute</i> Class can contain only an <i>XSDsimpleType</i> Class as its child Class. When such a Class is present as its child, all its inheritance is ignored. This Class can inherit from only one <i>XSDsimpleType</i> Class.

«XSDunion»

UML Construct		Class
Description		Creates an <code><xs:union></code> construct which can act as a container for <i>XSDsimpleType</i> Class.
Toolbox Icon		 Union
Tagged Values		None
Constraints		An <i>XSDunion</i> Class can contain only <i>XSDsimpleType</i> as its child Class and can generalize from other <i>XSDsimpleType</i> Classes only. All the Classes that this Class generalizes become the members of the attribute <i>memberTypes</i> . This Class cannot have any attributes or associations.

«XSDattributeGroup»

UML Construct		Class
Description		Creates an <code><XSDattributeGroup></code> construct which can act as a container for a set of elements for stereotype <i>XSDattribute</i> .
Toolbox Icon		 Attribute Group

Tagged Values		None
Constraints		<p>An <i>XSDattributeGroup</i> Class can contain only elements of stereotype <i>XSDattribute</i> and can be associated only with other <i>XSDattributeGroup</i> Classes.</p> <p>Only <i>XSDcomplexType</i> Classes can associate with this Class.</p> <p>This Class cannot be inherited.</p>

Learn More:

- [UML Profile for XSD](#) (Online Resource)
- [W3C XML Schema](#) (Online Resource)
- [Using Profiles](#)^[1026]

13.1.5.2 XSD Datatypes Package

When modeling XSD constructs, it is often useful to have the XSD primitive types represented as UML elements. In this way user-defined types, for example, can reference the datatype elements as part of inheritance or association relationships.

Sparx Systems provides the set of primitive XSD data types as a UML package in the form of an XMI file. Each of the XSD primitive types is represented by a UML Class in a package named *XSDDatatypes*.

How To:

Step	Action	See Also
1.	Download the <i>XSDDatatypes</i> package using the link to the right. The file <i>XSDDatatypes.xml</i> is an XMI file.	XSDDatatypes Package (XML file)
2.	Use Enterprise Architect's XMI import facility, which is available via the Project Model Import/Export Import Package from XMI menu option.	XMI import ^[323]
3.	When the XMI import is complete, you have the UML package named <i>XSDDatatypes</i> in your model, from which you can drag and drop the relevant types as required.	

Notes:

- Import and use the XSD Datatypes into Enterprise Architect only if you are modeling the XSD model using the UML Profile for XSD (instead of using the XML Schema Toolbox)

13.2 Web Services - WSDL

Enterprise Architect enables rapid modeling, forward engineering and reverse engineering of **Web Service Definition Language 1.1** (WSDL), a key W3C XML technology.

WSDL support is critical for the development of a complete *Service Oriented Architecture* (SOA), and the coupling of UML 2.3 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization. This section explains how to use Enterprise Architect to model WSDL files.

Learn More:

- [W3C WSDL Specification](#) (Online Resource)
- [WSDL Model Structure](#)^[1611]
- [Model WSDL](#)^[1612]
- [WSDL Model Transformation](#)^[1325]
- [Generate WSDL 1.1](#)^[1621]
- [Import WSDL 1.1](#)^[1622]

13.2.1 Create WSDL 1.1 Model Structure

Enterprise Architect provides the **WSDL** page in the **Toolbox** to conveniently model WSDL documents. WSDLs, represented as Components marked with the stereotype *WSDL*, are contained in a *WSDLnamespace* stereotyped Package whose hierarchy represents the target WSDL namespace and its constituent *XSD Types*, *Messages*, *PortTypes*, *Bindings* and *Services*. A sample WSDL Package structure can be created in the **Project Browser**, using the **Namespace** icon from the **WSDL** of the **Toolbox**. You can use this package structure as a template for developing your WSDL.

Access: [View](#) | [Toolbox > More tools](#) | [WSDL](#)

Use to:

- Create a new **WSDL** Package structure that can be used as a starting point for developing your WSDL model.

How To:

Step	Action	See Also
1	Create a package in the Project Browser	
2	Click on the New Diagram icon in the Project Browser	Project Browser Toolbar ^[452]
3	Select UML Structural in the Select From field and Class in the Diagram Types field	
4	Click on the More tools option in the Toolbox and select WSDL from the list of Toolboxes	Diagram Toolbox ^[546]
5	Click on the Namespace icon from the Toolbox and drag it into the created Class diagram	WSDL Group ^[565]
6	Specify the WSDL Package Name and Namespace and press OK button to create the sample WSDL Package structure.	Namespace Icon ^[1612]

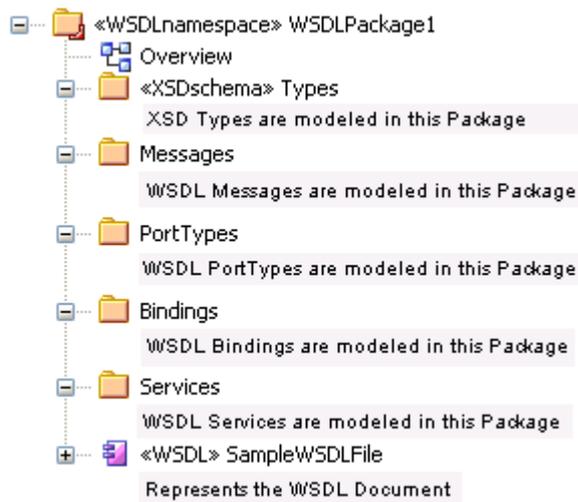
Alternative:

A WSDL Package structure can also be created from an UML Interface using the **WSDL Model Transformation**.

[WSDL Model Transformation](#)^[1325]

Example WSDL Package Structure:

The *WSDLnamespace* Package acts as a container for the WSDL Structure in Enterprise Architect. The various WSDL constructs like *Messages*, *PortTypes*, *Bindings* and *Services* should be modeled in the **Messages**, **PortTypes**, **Bindings** and **Services** Packages respectively.

**Notes:**

- A *WSDLnamespace* package can contain one or more *WSDL* stereotyped Component
- A *WSDL* stereotyped Component can be generated as a WSDL file

Learn More:

- [WSDL Toolbox](#)^[565]
- [Model WSDL](#)^[1612]
- [Generate WSDL 1.1](#)^[1621]
- [Import WSDL 1.1](#)^[1622]

13.2.2 Model WSDL

The **Namespace** icon in the **WSDL** Toolbox creates a *WSDLnamespace* stereotyped Package which represents the top-level container for the WSDL constructs in Enterprise Architect.

WSDL Properties: Double-click on the *WSDLnamespace* stereotyped package in the diagram or the Project Browser

Reference:

Field/Button	Usage	See Also
--------------	-------	----------

WSDL Package Name	Specify the name for the WSDL package	
Target Namespace	Specify the namespace for this WSDL package (<i>Optional</i>)	
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the Schema package.	Properties 

Learn More:

The *WSDLnamespace* stereotyped Package acts as a container for the following WSDL constructs:

- [WSDL Message](#) 
- [WSDL Message Part](#) 
- [WSDL Port Type](#) 
- [WSDL Port Type Operation](#) 
- [WSDL Binding](#) 
- [WSDL Binding Operation](#) 
- [WSDL Service](#) 
- [WSDL Document](#) 

13.2.2.1 WSDL Message

The **Message** icon in the **WSDL** Toolbox creates a *WSDLmessage* stereotyped Class, representing the WSDL Message. The *WSDLmessage* stereotyped Class acts as a container for one or more WSDL Message Parts. WSDL Message should be defined under the *Messages* package in the WSDL Package structure.

WSDL Properties: Double-click on the *WSDLmessage* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the WSDL Message	
Documentation	Specify the notes for this element (<i>Optional</i>)	
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the WSDL Message	Properties 

Notes:

- WSDL Messages can only be created under the *Messages* package in the WSDL Package structure
- The name of the WSDL Message should be unique among all the WSDL Messages within the WSDL

Learn More:

- [Message Part](#) ^[1614]
- [WSDL Package Structure](#) ^[1617]

13.2.2.1.1 WSDL Message Part

The **Message Part** icon from the **WSDL** Toolbox creates an UML Attribute when dropped on a *WSDLmessage* stereotyped Class. In Enterprise Architect, the WSDL Message Part must be associated with an **XSD** Type.

WSDL Properties: Double-click on the UML Attribute in the *WSDLmessage* stereotyped Class in the diagram or the Project Browser

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the WSDL Message Part	
Type	Specify : <ul style="list-style-type: none"> • XSD <i>built-in dataType</i> using the drop-down list or • XSD <i>Element</i> or <i>Complex Type</i> or <i>SimpleType</i> Class The XSD Types can be selected from : <ul style="list-style-type: none"> • Types package under the WSDL Package Structure or • Any package in the Project Browser 	XML Schema ^[1581] WSDL Package Structure ^[1611]
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the WSDL Message Part.	Properties ^[696]

Notes:

- *WSDLmessage* stereotyped Classes can contain WSDL Message Parts only

Learn More:

- [WSDL Message](#) ^[1613]

13.2.2.2 WSDL Port Type

The **Port Type** icon in the **WSDL** Toolbox creates a *WSDLportType* stereotyped Interface, representing the WSDL PortType. The *WSDLportType* stereotyped Interface describes the operations exposed by the WSDL and acts as a container for one or more WSDL PortType Operations. WSDL PortType should be defined under the *PortTypes* package in the WSDL Package structure.

WSDL Properties: Double-click on the *WSDLportType* stereotyped Interface in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the WSDL PortType	
Documentation	Specify the notes for this element (<i>Optional</i>)	
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the WSDL PortType	Properties ^[660]

Notes:

- WSDL PortTypes can only be created under the *PortTypes* package in the WSDL Package structure
- The name of the WSDL PortType should be unique among all the WSDL PortTypes within the WSDL

Learn More:

- [WSDL PortType Operation](#) ^[1615]
- [WSDL Package Structure](#) ^[1611]

13.2.2.2.1 WSDL Port Type Operation

The **PortType Operation** icon from the **WSDL** Toolbox creates an UML Operation when dropped on a *WSDLportType* stereotyped Interface. Enterprise Architect supports creating the following PortType Operations:

- *One-Way*
- *Request-Response*
- *Solicit-Response*
- *Notification*

WSDL Properties: Double-click on the UML Operation in the *WSDLportType* stereotyped Interface in the diagram or the Project Browser

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the WSDL PortType Operation	
Documentation	Specify the notes for this element (<i>Optional</i>)	
Operation Type	Select one of the supported PortType Operation types. <ul style="list-style-type: none"> • The Output section is disabled for <i>One-Way</i> Operation Type • The Input section is disabled for <i>Notification</i> Operation Type 	

Input	Specify the details of the input WSDL Message.		
	Field	Usage	
	Name	Specify the name of the input Message Defaults to the Operation Type .	
	Message	Select a <i>WSDLmessage</i> stereotyped C The list displays all the WSDL Message the <i>Message</i> package	Message ^[1613] WSDL Package Structure ^[1611]
	Documentation	Specify the notes for this input Message	
Output	Specify the details of the output WSDL Message.		
	Field	Usage	
	Name	Specify the name of the output Message Defaults to the Operation Type .	
	Message	Select a <i>WSDLmessage</i> stereotyped C The list displays all the WSDL Message the <i>Message</i> package	
	Documentation	Specify the notes for this input Message	
Faults	Specify the details of the WSDL Message that can act as Faults.		
	Button	Usage	
	New	Add a new Fault	
	Delete	Delete an existing Fault in the list	
OK	Save the values entered and close the dialog		
Cancel	Discard the values entered and close the dialog		
Help	Display this Help topic		
UML	Open the UML Properties dialog This button is available when editing the WSDL PortType Operation.		Properties ^[708]

Notes:

- WSDL PortType Operations can only be contained by WSDL PortTypes
- The value specified in the **Name** field of the **Input**, **Output** and **Fault** section above must be unique among all the Inputs, Outputs and Faults of the WSDL PortType respectively

Learn More:

- [WSDL Port Type](#)^[1613]

13.2.2.3 WSDL Binding

The **Binding** icon in the **WSDL** Toolbox creates a *WSDLbinding* stereotyped Class, representing the WSDL Binding. Each *WSDLbinding* Class implements the operations specified by a particular *WSDLportType* Interface and describes the message format and protocol details for the operations and messages defined by this WSDL PortType. WSDL Bindings should be defined under the *Bindings* package in the WSDL Package structure.

WSDL Properties: Double-click on the *WSDLbinding* stereotyped Class in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also
Name	Specify the name for the WSDL PortType	
PortType	Select the WSDL PortType to be implemented by this WSDL Binding.	Port Type ^[1614]
Protocol	Specify the protocol for the transmission of the selected WSDL PortType's Operations. The supported protocols are: <ul style="list-style-type: none"> • SOAP • HTTP 	
Transport	Defines the SOAP protocol. This field is disabled for HTTP protocol. Default: <code>http://schemas.xmlsoap.org/soap/http</code>	
Style	Specify the style of SOAP protocol. This field is disabled for HTTP protocol.	
Verb	Specify the HTTP verb. The supported verbs are: <ul style="list-style-type: none"> • GET • POST This field is disabled for SOAP protocol.	
Documentation	Specify the notes for this element (<i>Optional</i>)	
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the WSDL Binding.	Properties ^[660]

Notes:

- A WSDL Binding must implement a WSDL PortType. Therefore, WSDL PortTypes should be defined before creating WSDL Bindings
- WSDL Bindings can only be created under the *Bindings* package in the WSDL Package structure.

- The name of the WSDL Binding should be unique among all the WSDL Bindings within the WSDL.

Learn More:

- [WSDL Binding Operation](#) ^[1618]
- [WSDL Package Structure](#) ^[1619]

13.2.2.3.1 WSDL Binding Operation

When a *WSDLbinding* stereotyped Class is created using the WSDL Binding dialog, Enterprise Architect draws a Realization connector from the WSDL Binding to the WSDL Porttype and automatically populates the WSDL Binding with all the UML Operations in the WSDL PortType.

WSDL Properties: Double-click on the UML Operation in the WSDLbinding stereotyped Class in the diagram or the Project Browser

Reference:

Field/Button	Usage	See Also
Operation Name	Name of the Operation. This value in this field cannot be edited.	
Action	Specify the SOAP Action header (<i>URI</i>) for this Operation. This field is disabled for <i>HTTP</i> protocol.	
Style	Specify the style of this Operation. This field is disabled for <i>HTTP</i> protocol.	
Location	Specify the relative URI of this Operation . This field is disabled for SOAP protocol.	
Documentation	Specify the notes for this element (<i>Optional</i>)	
Parameters	Specify the details for the Operation Parameters. Click on the Details button to specify the details for Input, Output and Faults.	
	Field	Usage
	Use	Specify the encoding that is to be used. This field is disabled for <i>HTTP</i> protocol.
	Encoding Style	Specify the style (<i>URIs</i>) when Use is set to <i>enc</i> This field is disabled for <i>HTTP</i> protocol.
	Namespace	Specify the namespace (<i>Optional</i>) This field is disabled for <i>HTTP</i> protocol.
	Parts	Specifies the Message Parts that appear within the Body portion (<i>Optional</i>) This field is disabled for <i>HTTP</i> protocol.
	Header	Specify the contents of the SOAP/HTTP Header ()
	Documentation	Specify the notes for this input Message (<i>Optional</i>)

	<ul style="list-style-type: none"> • Details in Input section is disabled for <i>Notification</i> Operation Type. • Details in Output section is disabled for <i>One-way</i> Operation Type. • Details in Fault section is disabled where there are no Fault Messages. 	
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the WSDL Binding Operation.	Properties [708]

Learn More:

- [WSDL Binding](#) [1617]

13.2.2.4 WSDL Service

The **Service** icon in the **WSDL** Toolbox creates a *WSDLservice* stereotyped Interface, representing the WSDL Service. The *WSDLservice* stereotyped Interface describes the collection of ports that expose a particular Binding. WSDL Service should be defined under the *Services* package in the WSDL Package structure.

WSDL Properties: Double-click on the *WSDLservice* stereotyped Interface in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also																
Name	Specify the name for the WSDL Service																	
Documentation	Specify the notes for this element (<i>Optional</i>)																	
Ports	Specify the Ports (or endpoints) for this WSDL Service. <table border="1"> <thead> <tr> <th>Button</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td colspan="2">Click on the New button to add a Port using the WSDL Port dialog</td> </tr> <tr> <th>Field</th> <th>Usage</th> </tr> <tr> <td>Port Name</td> <td>Specify the name of the Port</td> </tr> <tr> <td>Binding</td> <td>Select a <i>WSDLbinding</i> stereotyped Class The list displays all the WSDL Bindings of the <i>Bindings</i> package</td> </tr> <tr> <td>Location</td> <td>Specify the URI for the Port</td> </tr> <tr> <td>Documentation</td> <td>Specify the notes for this input Message</td> </tr> <tr> <td>OK</td> <td>Save the values entered and close the W</td> </tr> </tbody> </table>	Button	Usage	Click on the New button to add a Port using the WSDL Port dialog		Field	Usage	Port Name	Specify the name of the Port	Binding	Select a <i>WSDLbinding</i> stereotyped Class The list displays all the WSDL Bindings of the <i>Bindings</i> package	Location	Specify the URI for the Port	Documentation	Specify the notes for this input Message	OK	Save the values entered and close the W	Binding [1617]
Button	Usage																	
Click on the New button to add a Port using the WSDL Port dialog																		
Field	Usage																	
Port Name	Specify the name of the Port																	
Binding	Select a <i>WSDLbinding</i> stereotyped Class The list displays all the WSDL Bindings of the <i>Bindings</i> package																	
Location	Specify the URI for the Port																	
Documentation	Specify the notes for this input Message																	
OK	Save the values entered and close the W																	

	<table border="1"> <tr> <td>Cancel</td> <td>Discard the values entered and close the dialog</td> </tr> <tr> <td>Help</td> <td>Display this Help topic</td> </tr> <tr> <td colspan="2">Press the Delete button to remove an entry from the list</td> </tr> </table>	Cancel	Discard the values entered and close the dialog	Help	Display this Help topic	Press the Delete button to remove an entry from the list		
Cancel	Discard the values entered and close the dialog							
Help	Display this Help topic							
Press the Delete button to remove an entry from the list								
OK	Save the values entered and close the dialog							
Cancel	Discard the values entered and close the dialog							
Help	Display this Help topic							
UML	Open the UML Properties dialog This button is available when editing the WSDL Service.	Properties ^[660]						

Notes:

- WSDL Services can only be created under the *Service* package in the WSDL Package structure.
- The name of the WSDL Service should be unique among all the WSDL Services within the WSDL.

Learn More:

- [WSDL Package Structure](#) ^[161]

13.2.2.5 WSDL Document

The **WSDL** icon in the **WSDL** Toolbox creates a *WSDL* stereotyped Component, representing the WSDL document. The *WSDL* stereotyped Component should be modeled as direct child element of the *WSDLnamespace* stereotyped Package. Multiple WSDL documents can be created for a single *WSDLnamespace* stereotyped Package, thus enabling the WSDL Services for that namespace to be reused and exposed as required across multiple WSDLs.

WSDL Properties: Double-click on the *WSDL* stereotyped Component in the diagram or the **Project Browser**

Reference:

Field/Button	Usage	See Also					
Name	Specify the name for the WSDL document						
File Name	Specify the file path where the WSDL 1.1 file is to be generated.						
Documentation	Specify the notes for this element (<i>Optional</i>)						
XMLNS	List specifying the additional namespace and namespace-prefix pairs used in this WSDL document.						
	<table border="1"> <tr> <td>Button</td> <td>Usage</td> </tr> <tr> <td colspan="2">Press the New button to add an entry in the list.</td> </tr> <tr> <td colspan="2">Double-click on an entry in the list to edit the values.</td> </tr> </table>	Button	Usage	Press the New button to add an entry in the list.		Double-click on an entry in the list to edit the values.	
Button	Usage						
Press the New button to add an entry in the list.							
Double-click on an entry in the list to edit the values.							

Field	Usage	
Prefix	Specify the abbreviated value that represents the N	
Namespace	Specify the namespace	
OK	Save the values entered and close the Namespace	
Cancel	Discard the values entered and close the Namesp	
Help	Display this Help topic	
Press the Delete button to remove an entry from the list		
Services	Select the required WSDL Services. The list displays all the WSDL Services created in the <i>Services</i> package	
OK	Save the values entered and close the dialog	
Cancel	Discard the values entered and close the dialog	
Help	Display this Help topic	
UML	Open the UML Properties dialog This button is available when editing the WSDL document.	Properties [660]

13.2.3 Generate WSDL

The **Generate WSDL** feature forward engineers a UML model to a WSDL 1.1 file. The **Generate WSDL** feature acts either on a *WSDLnamespace* stereotyped Package or a *WSDL* stereotyped Component. It is used to generate any or all of the *WSDL* stereotyped Components owned by the target *WSDLnamespace* structure.

Access:

- **Tools | Web Service | Generate WSDL**
- Right-click on a *WSDL* stereotyped Component and select the following context menu : **Generate WSDL**

Reference:

Field/Button	Usage	See Also				
Encoding	Specify the encoding scheme for the WSDL file. Default: Value of the Code Page field of the XML Specifications section in the Options dialog	XML Specifications [436]				
Select Components To Generate	Lists all the <i>WSDL</i> stereotyped Components under the current <i>WSDLnamespace</i> Package. Highlight the required <i>WSDL</i> stereotyped Components for which WSDL file is to be generated.					
	<table border="1"> <thead> <tr> <th>Button</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Button	Usage			
Button	Usage					

Field/Button	Usage	See Also												
	<table border="1"> <tr> <td>Select All</td> <td>Selects all the entries in this list</td> </tr> <tr> <td>Select None</td> <td>Selects none the entries in this list</td> </tr> </table> <p>Double-click on an entry in the list to open the Component File Name dialog:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Specifies the name of the <i>WSDL</i> stereotyped Comp</td> </tr> <tr> <td>Prefix</td> <td>Select the desired prefix for the WSDL namespace if prefixes had been specified</td> </tr> <tr> <td>File Name</td> <td>Specify the file path where the WSDL file is to be ge</td> </tr> </tbody> </table>	Select All	Selects all the entries in this list	Select None	Selects none the entries in this list	Field	Usage	Name	Specifies the name of the <i>WSDL</i> stereotyped Comp	Prefix	Select the desired prefix for the WSDL namespace if prefixes had been specified	File Name	Specify the file path where the WSDL file is to be ge	
Select All	Selects all the entries in this list													
Select None	Selects none the entries in this list													
Field	Usage													
Name	Specifies the name of the <i>WSDL</i> stereotyped Comp													
Prefix	Select the desired prefix for the WSDL namespace if prefixes had been specified													
File Name	Specify the file path where the WSDL file is to be ge													
Generate	Generate the WSDL file for all the selected <i>WSDL</i> stereotyped Components													
View WSDL	View the generated WSDL for a selected <i>WSDL</i> stereotyped Component													
Close	Close this dialog													
Help	Display this Help topic													
Progress	Lists the progress of WSDL Generation													

13.2.4 Import WSDL

Enterprise Architect provides a **WSDL Import** facility to reverse engineer WSDL 1.1 files into UML Class models.

Access:

- **Tools | Web Service | Import WSDL**
- Right-click on a Package in the **Project Browser** and select the following context menu : **Code Engineering | Import WSDL**

Reference:

Field/Button	Usage
Filename	Specify the WSDL file to import
Root Package	Specifies the Package under which the WSDL file is to be imported
Target Package	Specifies the name of the Package that represents the imported WSDL file Default: Name of the WSDL file
Import	Initiate the WSDL Import
Close	Close this dialog

Field/Button	Usage
Help	Display this Help topic
Progress	Lists the progress of WSDL Import

Notes:

Enterprise Architect cannot import a WSDL file that references WSDL constructs existing outside this WSDL file	
Importable Example	http://www.w3.org/TR/wsdl.html#_wsdl
Non-importable Example	http://www.w3.org/TR/wsdl.html#_style Attempting to import the second WSDL file would result in the following error: Cannot Import Split Files
Workaround	Combine the split WSDL files into a single file and then import it into Enterprise Architect

13.3 SoaML

Topics:

Topic	Detail	See also
Abstract	<p>The following text is derived from Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and metamodel for Services (UPMS) (OMG document ad/2008-11-01); pp. 25-26:</p> <p><i>A service is an offer of value to another through a well-defined interface and available to a community (which may be the general public). A service results in work provided to one by another.</i></p> <p><i>Service Oriented Architecture (SOA) is a way of organizing and understanding (representations of) organizations, communities and systems to maximize agility, scale and interoperability. The SOA approach is simple - people, organizations and systems provide services to each other. These services allow us to get something done without doing it ourselves or even without knowing how to do it - enabling us to be more efficient and agile. Services also enable us to offer our capabilities to others in exchange for some value - thus establishing a community, process or marketplace. The SOA paradigm works equally well for integrating existing capabilities as for creating and integrating new capabilities.</i></p> <p><i>SOA ... is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results. SoaML ... provides a standard way to architect and model SOA solutions using the Unified Modeling Language (UML). The profile uses the built-in extension mechanisms of UML to define SOA concepts in terms of existing UML concepts.</i></p> <p><i>... the highest leverage of employing SOA comes from understanding a community, process or enterprise as a set of interrelated services and ... supporting that service oriented enterprise with service-enabled systems. SoaML enables business oriented and systems oriented services architectures to mutually and collaboratively support the enterprise mission. ... SoaML depends on Model Driven Architecture® (MDA®) to help map business and systems architectures, the design of the enterprise, to the technologies that support SOA, like web services and CORBA®.</i></p> <p>For further information on the concepts of SoaML, see the specification document on the OMG website SOAML document page.</p>	<p>SOAML document page (Online Resource)</p>
SoaML in Enterprise Architect	<p>Enterprise Architect enables you to model services architectures quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The SoaML facilities are provided in the form of:</p> <ul style="list-style-type: none"> • Two SoaML diagram types - <i>SoaML Component Diagram</i> and <i>SoaML Sequence Diagram</i> - accessed through the New Diagram dialog • SoaML pages in the Toolbox • SoaML element and relationship entries in the Toolbox Shortcut Menu and Quick Linker 	<p>New Diagram [568]</p> <p>Toolbox Shortcut Menu [551]</p> <p>Quick Linker [622]</p>
SoaML Toolbox Pages	<p>You can access the SoaML pages of the Toolbox through the More tools SoaML menu option. There is a set of pages for each SoaML</p>	

Topic	Detail	See also
	diagram type, although the last five pages in each set are the same.	

Topic	Detail	See also
	<p>SoaML Component Diagram</p> <ul style="list-style-type: none"> UML Component <ul style="list-style-type: none"> Package Packaging Component Component Class Interface Object Part Expose Interface UML Component Relationships <ul style="list-style-type: none"> Assembly Delegate Associate Realize Generalize Nesting SOA - Contracts <ul style="list-style-type: none"> Collaboration Service Contract Services Architecture Participant Architecture... Collaboration Use SOA - Services <ul style="list-style-type: none"> Agent... Entity ID... Participant... Service Interface... Specification Port Request Point Service Point Service Channel SOA - Service Data <ul style="list-style-type: none"> Attachment... Message Type SOA - Milestones <ul style="list-style-type: none"> Milestone SOA - Capabilities <ul style="list-style-type: none"> Capability... Expose 	<p>SoaML Sequence Diagram</p> <ul style="list-style-type: none"> UML Sequence <ul style="list-style-type: none"> Lifeline Fragment Endpoint Diagram Gate State/Continuation UML Sequence <ul style="list-style-type: none"> Message Self-Message Call Recursion SOA - Contracts SOA - Services SOA - Service Data SOA - Milestones SOA - Capabilities Common

Topic	Detail	See also
<p>Example SoaML Diagram</p>	<p>The diagram illustrates a SoaML services architecture for a dealer network. At the top level, a dashed oval represents the «servicesArchitecture» named DealerNetworkArchitecture. Inside this architecture, a «collaboration» named ps :PurchasingService is defined. This collaboration has two roles: 'buyer' and 'seller'. The 'buyer' role is associated with a participant «participant» dealer, and the 'seller' role is associated with a participant «participant» mfg. Below these participants, there are detailed boxes for «participant» Dealer and «participant» Manufacturer. The Dealer participant has a «servicePoint» Buyer, and the Manufacturer participant has a «servicePoint» ServicePoint. A «Comment» box provides context: 'The purpose of the dealer network architecture is to establish the financially related services between dealers and manufacturers such that any dealer can do business with any manufacturer.' Dashed arrows labeled 'type' connect the participant boxes to their respective detailed boxes.</p>	
<p>Disable SoaML</p>	<p>If you prefer not to use SoaML in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies).</p>	<p>Manage MDG Technologies 1033</p>

Notes:

- Service Oriented Architecture Modeling Language (SoaML) is supported in the Corporate, Systems Engineering, Business and Software Engineering and Ultimate editions of Enterprise Architect

13.4 SOMF

Topics:

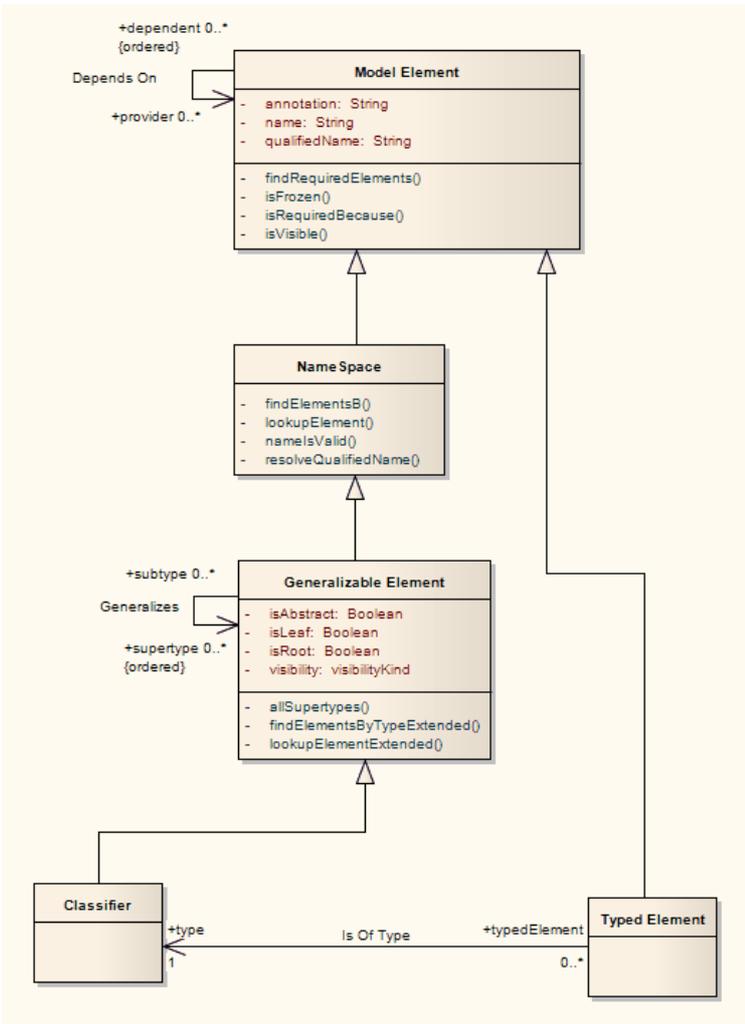
Topic	Detail	See also
Abstract	<p>The following text is derived from the extensive Wikipedia entry on Service Oriented Modeling:</p> <p><i>The Service-Oriented Modeling Framework (SOMF) has been proposed by author Michael Bell as a holistic and anthropomorphic modeling language for software development that employs disciplines and a universal language to provide tactical and strategic solutions to enterprise problems. The term "holistic language" pertains to a modeling language that can be employed to design any application, business and technological environment, either local or distributed. This universality may include design of application-level and enterprise-level solutions, including SOA landscapes or Cloud Computing environments. The term "anthropomorphic", on the other hand, affiliates the SOMF language with intuitiveness of implementation and simplicity of usage.</i></p> <p><i>The service-oriented modeling framework (SOMF) is a service-oriented development life cycle methodology. It offers a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle management and modeling. ... It illustrates the major elements that identify the "what to do" aspects of a service development scheme. These are the modeling pillars that will enable practitioners to craft an effective project plan and to identify the milestones of a service-oriented initiative- either a small or large-scale business or a technological venture.</i></p> <p>For further information on the concepts of SOMF, see the Wikipedia Service Oriented Modeling pages.</p>	<p>Michael Bell (Online Resource)</p> <p>Service Oriented Modeling (Online Resource)</p>
SOMF in Enterprise Architect	<p>In Enterprise Architect, SOMF is implemented as a profile within an MDG Technology that is integrated with the Enterprise Architect installer. For further information on this MDG Technology, see the <i>Service-Oriented Modeling Framework™ in Enterprise Architect</i> page on the Sparx Systems website.</p> <p>The SOMF facilities are provided in the form of:</p> <ul style="list-style-type: none"> Eight SOMF diagram types accessed through the New Diagram dialog; these are SOMF: <ul style="list-style-type: none"> <i>Analysis</i> <i>Business Integration</i> <i>Logical Design Relationship</i> <i>Logical Design Composition</i> <i>Transaction</i> <i>Conceptual Architecture</i> <i>Asset Utilization</i> <i>Transaction Directory</i> SOMF pages in the Toolbox SOMF element and relationship entries in the Toolbox Shortcut menu and Quick Linker 	<p>Sparx Systems (online resource)</p> <p>New Diagram <small>[568]</small></p> <p>Toolbox Shortcut Menu <small>[55↑]</small></p> <p>Quick Linker <small>[62↓]</small></p>
SOMF Toolbox Pages	<p>You can access the SOMF pages of the Toolbox through the More tools SOMF menu option. There is a set of pages for each SOMF diagram type; for example:</p>	

Topic	Detail	See also
	<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p>SOMF Conceptual Architecture Diagram</p> <ul style="list-style-type: none"> [-] SOMF - Conceptual Architecture Assets <ul style="list-style-type: none"> Architectural Concept Business Domain Packaged Technological Asset Technological Function [-] SOMF - Conceptual Architecture Operations <ul style="list-style-type: none"> Recognized As Conceptualized As Function Of Extended Owner Of </div> <div style="width: 30%;"> <p>SOMF Business Integration Diagram</p> <ul style="list-style-type: none"> [-] SOMF - Business Integration Assets <ul style="list-style-type: none"> Business Tier Business Domain Contextual Perspective Atomic Service Composite Service Service Cluster [-] SOMF - Business Integration Operations <ul style="list-style-type: none"> Integrated Disintegrated Contained Separated Perspective Of </div> <div style="width: 30%;"> <p>SOMF Transaction Diagram</p> <ul style="list-style-type: none"> [-] SOMF - Transaction Assets <ul style="list-style-type: none"> Actor Lifeline Atomic Service Composite Service Service Cluster Consumer Consumer_Service Section Aggregator Entity Section Session Section Transaction Section Activity Section [-] SOMF - Transaction Activity Connectors <ul style="list-style-type: none"> Originating Activity Connector Intermediating Activity Connector End-of-Activity Connector </div> </div>	
Disable SOMF	If you prefer not to use SOMF in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (Settings MDG Technologies)	MDG Technologies <small>1033</small>

13.5 Generate MOF

Enterprise Architect offers support for exporting packages to XML under the *Meta-Object Facility (MOF)* 1.3 and 1.4 standards. MOF models are created by assigning the stereotype *metamodel* to the package. MOF models can be exported to MOF 1.3 or MOF 1.4 XML file specification.

Topics:

Topic	Detail	See also
<p>Background Knowledge</p>	<p>MOF is an Object Management Group (OMG) standard that originated in the UML, when the OMG required a Meta-Modeling architecture to define the UML. MOF is designed as a four-layered architecture, as illustrated in the following diagram.</p>  <pre> classDiagram class ModelElement { +annotation: String +name: String +qualifiedName: String -findRequiredElements() -isFrozen() -isRequiredBecause() -isVisible() } class Namespace { -findElementsB() -lookupElement() -nameIsValid() -resolveQualifiedName() } class GeneralizableElement { +isAbstract: Boolean +isLeaf: Boolean +isRoot: Boolean +visibility: visibilityKind -allSupertypes() -findElementsByTypeExtended() -lookupElementExtended() } class Classifier class TypedElement ModelElement < -- Namespace ModelElement < -- GeneralizableElement GeneralizableElement < -- Classifier GeneralizableElement < -- TypedElement ModelElement "0..*" -- "0..*" ModelElement : Depends On ModelElement "0..*" -- "0..*" ModelElement : +provider GeneralizableElement "0..*" -- "0..*" GeneralizableElement : +subtype GeneralizableElement "0..*" -- "0..*" GeneralizableElement : +supertype Classifier "1" -- "0..*" TypedElement : +type TypedElement "0..*" -- "1" Classifier : Is Of Type TypedElement "0..*" -- "0..*" ModelElement : +typedElement </pre> <p>Because of the similarities between the MOF-model and UML structure models, MOF meta-models are usually modeled as UML Class diagrams. You can also use the Metamodel page of the Toolbox to create MOF model elements and connectors. A supporting standard of MOF is XML, which defines an XML-based exchange format.</p> <p>MOF is a closed, strict meta-modeling architecture; every model element on every layer is strictly an instance of a model element of the layer above. MOF only provides a means to define the structure</p>	<p>Class Diagrams ⁷⁹⁸</p> <p>Metamodel Group ⁵⁶</p>

Topic	Detail	See also
	<p>or abstract syntax of a languages or of data.</p> <p>Simplified, MOF uses the notion of Classes, as known from object orientation, to define concepts (model elements) on a meta-layer. These Classes (concepts) can then be instantiated through objects (instances) of the model layer below. Because an element on the M2 layer is an object (instance of an M3 model element) as well as a Class (an M2 layer concept) the notion of a <i>clabject</i> is used. <i>Clabject</i> is a merge of the words Class and Object.</p> <p>Another related standard is OCL, which describes a formal language that can be used to define model constraints by means of predicate logic.</p>	

Learn More:

- [Getting Started](#)¹⁶³¹
- [Export MOF to XML](#)¹⁶³²

13.5.1 Getting Started

MOF diagrams are **Class** diagrams that are contained in packages with a metamodel stereotype.

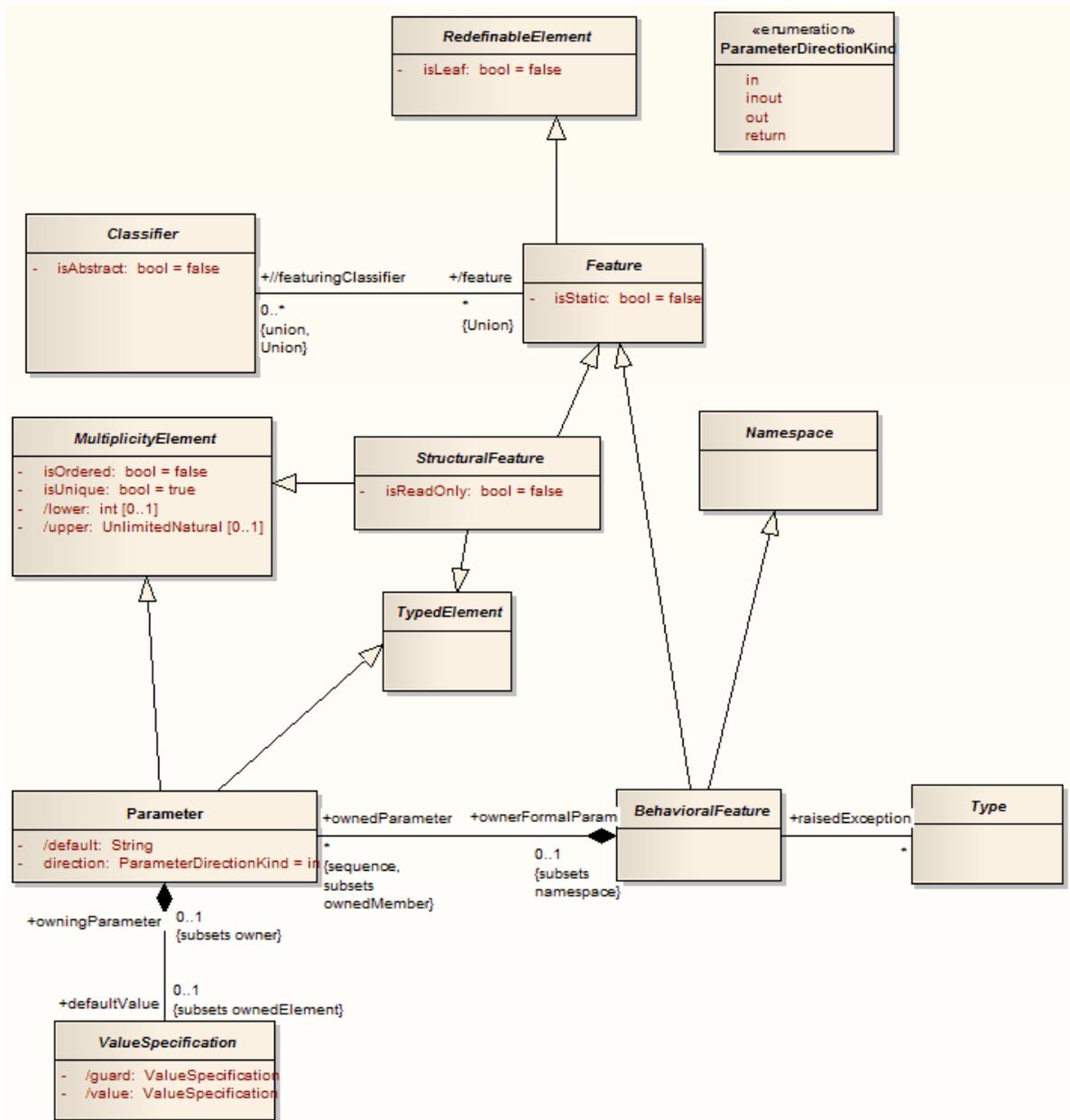
To create a MOF diagram, follow the steps below

How To:

Step	Action	See Also
1.	Create a package to contain your MOF elements.	
2.	Double-click on the package name to display the Properties dialog.	
3.	In the Stereotype field type the value metamodel .	
4.	Click on the OK button.	
5.	Right-click on the package in the Project Browser and select the Add Add Diagram context menu option. Create a Class diagram (the default diagram).	
6.	Give your MOF Class diagram an appropriate name.	
7.	In the Toolbox , select the More tools Metamodel menu option and add the required Metamodel elements to the diagram.	Metamodel Group ¹⁵⁶¹

Example:

The following is an example of a MOF diagram. A MOF diagram can typically contain Package, Class, Enumeration and Primitive elements, and Generalization, Association, Compose and Aggregate relationships.

**Learn More:**

- [Class Diagram](#) ^[798]

13.5.2 Export MOF to XML

To export a MOF diagram to XML, specifying the MOF 1.3 or MOF 1.4 standard, follow the steps below

How To:

Step	Action	See Also
1.	Right-click on the package in the Project Browser . The context menu displays.	

Step	Action	See Also
2.	Select the Import/Export Export Package to XMI file menu option. The Export Package to XMI dialog displays:	
3.	In the Filename field, type a name for the XMI file.	
4.	De-select the Enable full EA Roundtrip checkbox.	
5.	In the XMI Type field, click on the drop-down arrow and select MOF 1.3 or MOF 1.4 .	
6.	Click on the Export button and wait until the Progress bar reads 100% .	
7.	Once your file has been created, you can view it by clicking on the View XMI button.	Metamodel Group [56↑]

Notes:

- MOF diagrams exported to XMI can be imported using the regular import XMI features of Enterprise Architect. See **Import from XMI**

Learn More:

- [Import from XMI](#) [323]

Part



14 Visual Execution Analyzer



This section describes the recording and visual analysis of executing applications in Enterprise Architect.

The Visual Execution Analyzer builds on and processes the structures and operations of the Model Driven Development Environment (MDDE). The MDDE provides tools to design, build and debug an application, forming part of the Debugger facilities of Enterprise Architect.

One of the primary objectives of the Visual Execution Analysis feature is to enable you to work with the stack traces you captured in debugging your application, to generate Sequence diagrams. This is a great way to document and understand what your program is doing during its execution phase.

Access: **Analyzer**
Project Manager package context menu | Execution Analyzer

Use To:

- Optimize existing system resources and understand resource allocation
- Ensure that the system is following the rules as designed
- Produce high quality documentation that more accurately reflects system behavior
- Understand how and why systems work
- Train new employees in the structure and function of a system
- Provide a comprehensive understanding of how existing code works
- Identify costly or unnecessary function calls
- Illustrate interactions, data structures and important relationships within a system
- Trace problems to a specific line of code, system interaction or event
- Visualize why a sequence of events is important
- Establish the sequence of events that occur immediately prior to system failure
- Simulate the execution of behavior models including State Machines, Activities and Interactions

Topics:

Topic	See
Record executing programs and represent the behavior as a UML Sequence diagram; recording is supported for: <ul style="list-style-type: none"> • Microsoft Windows Native C • Microsoft Windows Native C++ • Microsoft Windows Visual Basic • Microsoft .NET Family (C#, J#, VB) • Sun Microsystems Java • PHP 	Recording Sequence Diagrams ^[1637]
Simulate UML behavior models to verify their logical and design correctness, for:	Model Simulation ^[1681]

Topic	See
<ul style="list-style-type: none">• Activities• Interactions and Sequences• State Machines	
Run xUnit tests	Unit Testing ^[1669]
Record native Windows applications and profile their behavior	Profiling Native Applications ^[1658]
Create and work with objects created within the Enterprise Architect modeling environment using a dynamic Object Workbench	Object Workbench ^[1663]
Testpoint Management - provides the facility to pass or fail application tasks, viewing test results in real time as the program executes and results are saved	TestPoint Management ^[1672]

Learn more:

- [Code, Build & Debug](#)^[1390] (for the Enterprise Architect MDDE and Debugger)

14.1 Recording Sequence Diagrams

This section explains how to use the Visual Execution Analyzer to record execution data in the form of a Sequence Diagram.

A Sequence diagram provides easy to understand visual information including:

- A representation of how information is passed through a system
- The sequence of various functions and their corresponding parameters
- A clear view of how different Classes interact to create behavior
- A visual overview of how data structures are used to produce results

A Sequence diagram extends traditional analysis to help identify errors in logic, explain unexpected system behavior and identify data flow inconsistencies; the Visual Execution Analyzer extends analysis through the use of a comprehensive array of reports that detail everything from state transitions through to the contents of the stack at a given time.

Access: [Analyzer](#) | [Record](#) | [Recorder](#)

Use To:

- Set up and control the recording of code execution
- Interpret the results of recording
- Generate Sequence diagrams from the recording, and moderate the contents of the diagram

Topics:

Review the following topics in sequence:

Topic	Link
An overview of how the Visual Execution Analyzer creates a visual representation of the execution of an application, outlining the functions that are being called, the types of messages being sent, the key data structures used and the relationships between different Classes The diagram makes it much simpler to understand how information is moved throughout the system and what values are being passed by various functions	How It works  ^[1638]
Preparing to record execution of the application	Setup for Recording  ^[1639]
Deploying recording markers, breakpoints and marker sets	Place Recording Markers  ^[1642]
Controlling the recording session, using the Record & Analyze window	Control the Recording Session  ^[1648]
Interpreting the results of a recording, saving and retrieving these results, generating a Sequence diagram from the recording, and reviewing the diagram	Generating Sequence Diagrams  ^[1651]
A description of how you can generate Sequence diagrams that show transitions in state as a program executes	Add State Transitions  ^[1653]

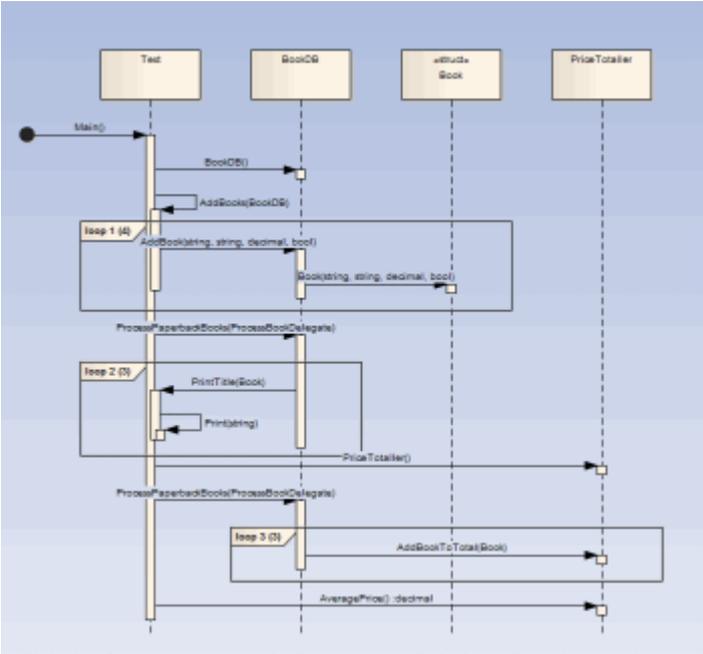
Notes:

- Recording is available to users of all editions of Enterprise Architect except the Desktop edition

14.1.1 How it Works

This topic explains how the Visual Execution Analyzer generates Sequence diagrams.

Topics:

Topic	Detail	See also
<p>Usage</p>	<p>The Visual Execution Analyzer enables you to generate a Sequence Diagram</p> <p>The diagram creates a visual representation of the execution of an application, outlining what functions are being called, the types of messages being sent, the key data structure used and the relationships between different Classes; that is, how information is moved throughout the system and what values are being passed by various functions</p> <p>The diagram below illustrates the Sequence Diagram output for a program that calculates the price of books</p> <p>The first loop structure is executed four times, to add four books to the book database; the arrows indicate information flow and demonstrate the change of states over time</p>  <p>Enterprise Architect can record arguments to functions, record calls to external modules or capture state transitions based on any given constraint; this information can be integrated with existing system knowledge and test data to optimize code execution, reduce errors and determine why application failure and system crashes occur</p> <p>If an application crashes, data corruption such as a stack overflow can prevent you from diagnosing and rectifying the problem; however, the Visual Execution Analyzer enables you to record a given execution sequence to provide a reliable source of information that</p>	

Topic	Detail	See also
	<p>might further explain why a crash occurred</p> <p>A Sequence diagram can convey more detail and provide greater understanding than reading unfamiliar code that might have been written by someone else; it also makes it easier to document existing code when the Sequence diagram illustrates functions that are being called and the specific sequence of events that occur to produce a particular type of system behavior</p>	

Learn more:

- [Setup for Recording](#)^[1639]

14.1.2 Setup for Recording

This section explains how you prepare to record execution of the application.

Topic	Link
Satisfying the prerequisites for recording execution of the application	Prerequisites ^[1639]
Controlling the detail of a recording by adjusting the stack depth and applying filters	Configuring Recording Detail ^[1639]

14.1.2.1 Pre-Requisites

This topic discusses the prerequisites for recording execution of the application

To set up the environment for recording Sequence diagrams you must

- Have completed the Basic setup for Build & Debug
- Be able to successfully debug the application

Learn More:

- [Build and Debug Prerequisites](#)^[1391]
- [Build and Debug Setup](#)^[1392]
- [Debugging](#)^[1416]

14.1.2.2 Configure Recording Detail

To avoid generating an overly complex sequence diagram, you need to control of the scope of what is being recorded.

To narrow the scope of the recording, you can configure two Analyzer properties:

- Filtering - using the Analyzer Script you can exclude specific modules, Classes and functions from the recording
- Stack Depth - this can be used to speed up recording and present a clearer diagram (see Notes below)

Access: **Analyzer | Execution Analyzer (Shift + F12) Double-click on required Analyzer Script > Recording**

Topics:

Topic	Detail	See also
Filtering	<p>If the Enable Filter checkbox is selected on the Recording page of the Execution Analyzer Script Editor, the debugger excludes calls to matching methods from the generated Sequence history and diagram; the comparison is case-sensitive</p> <p>To add a value, click on the New (Insert) icon in the right corner of the Exclusion Filters box, and type in the comparison string; each filter string takes the form:</p> <pre>class_name_token::method_name_token</pre> <p>The <i>class_name_token</i> excludes calls to all methods of a Class or Classes that have a name matching the token; the string can contain the wildcard character * (asterisk)</p> <p>The <i>method_name_token</i> excludes calls to methods having a name that matches the token; again, the string can contain the wildcard character *</p> <p>Both tokens are optional; if no Class token is present, the filter is applied only to global or public functions (that is, methods not belonging to any Class)</p>	Managing Analyzer Scripts ^[1392] Analyzer Script Editor ^[1394]
Example	<p>In the Java example below, the debugger would exclude:</p> <ul style="list-style-type: none"> • Calls to the <i>OnDraw</i> method for the Class <i>Example.common.draw.DrawPane</i> • Calls to any method of any Class having a name beginning with <i>Example.source.Collection</i> • Calls to any constructor for any Class (such as <i><clint></i> and <i><init></i>) <div data-bbox="427 1227 874 1377" style="border: 1px solid gray; padding: 5px;"> <p>Filters</p> <p>Example.common.draw.DrawPane::OnDraw</p> <p>Example.source.Collection*</p> <p>*:init*</p> </div> <p>In the Native Code example below, the debugger would exclude:</p> <ul style="list-style-type: none"> • Calls made to Standard Template Library namespace • Calls to any Class beginning with <i>TOb</i> • Calls to any method of Class <i>CLock</i> • Calls to any Global or Public Function with a name beginning with <i>Get</i> • Calls to the method <i>GetLocation</i> for Class <i>Ctrain</i> <div data-bbox="427 1675 874 1892" style="border: 1px solid gray; padding: 5px;"> <p>Filters</p> <p>std*</p> <p>TOb*</p> <p>CLock</p> <p>CTrain::GetLocation</p> <p>::Get*</p> </div>	

Reference:

To Filter	Use Filter Entry
All public functions having a name beginning with Get from the recording session (for example, <i>GetClientRect</i> in Windows API)	:: Get *
All methods beginning with Get for every Class member method	* :: Get *
All methods beginning with Get from the Class <i>CClass</i>	CClass :: Get *
All methods for Class <i>CClass</i>	CClass :: *
All methods for Classes belonging to Standard Template and Active Template Libraries	ATL* std*
The specific method <i>GetName</i> for Class <i>CClass</i>	CClass :: GetName

Notes:

By default, the Analyzer stack depth is restricted to four frames. This is relative to the depth at which the recording marker is encountered.

You can control stack depth on the toolbar of either

- The Breakpoint and Markers window or
- The Record & Analyze window

Learn More:

- [Control Stack Depth](#)^[1641]
- [Marker Types](#)^[1642]
- [Recorder Toolbar](#)^[1648]

14.1.2.3 Control Stack Depth

When recording particularly high-level points in an application, the stack frame count can result in a lot of information being collected; to achieve a quicker and clearer picture, it is better to limit the stack depth by setting the stack frame threshold.

Topics:

Topic	Detail	See also
Usage	Set the Control Stack depth in the numerical field on the toolbar of Breakpoints and Markers window or the Record & Analyze window	Breakpoint & Markers ^[1646]

Topic	Detail	See also
	 <p>The depth is relative to the stack depth where the Debugger first encountered the recording marker; so, when recording begins, if the stack depth is 6 and the limit depth is set to 3, the Debugger will record to a stack depth of 9</p> <p>For situations where the stack is very large, it is recommended that you first use a low stack frame threshold of 2 or 3 and gradually increase it if necessary to expand the picture.</p>	Control the Recording Session ^[1648]

14.1.3 Place Recording Markers

This section explains how to deploy recording markers:

Access: [Analyzer | Break Points and Markers](#)

Use To:

Trace marking enables you to silently record code execution between two points. This can be used for generating a Sequence diagram. As this records the execution of multiple threads, it can be particularly useful in capturing event driven sequences (such as mouse and timer events).

Topic	Link
Explaining the types and effects of the various kinds of recording markers, which are similar to breakpoints but determine the action the debugger takes when it comes to one	Marker Types ^[1642] Recording Activity for a Class ^[1646] Recording Activity for a Single Method ^[1645]
How to manage breakpoints in the Breakpoint & Markers window	The Breakpoint & Markers Window ^[1646]
How to activate and deactivate markers	Activate and Disable Markers ^[1647]
Working with Marker Sets	Working with Marker Sets ^[1648]

14.1.3.1 Marker Types

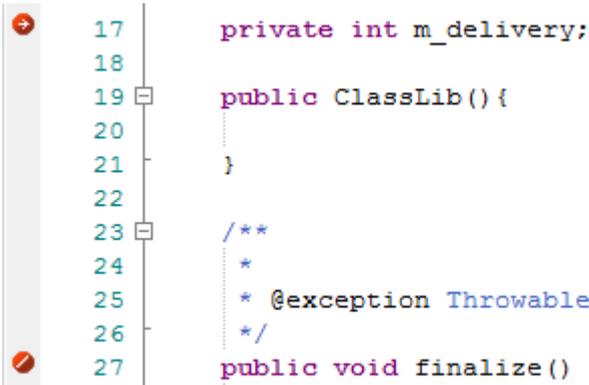
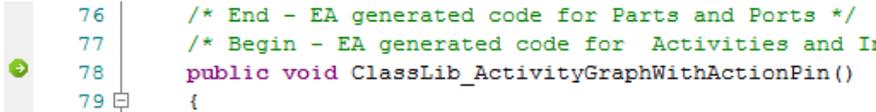
The recording markers are similar to breakpoints; however, instead of breaking execution as it does at a breakpoint, the debugger takes the action defined by the type of marker. If the marker is denoted as a recording *start point*, the debugger immediately begins to trace all executed calls from that point for the breaking thread. Recording is stopped again when either the thread that is being captured terminates or the thread encounters a recording *end point*.

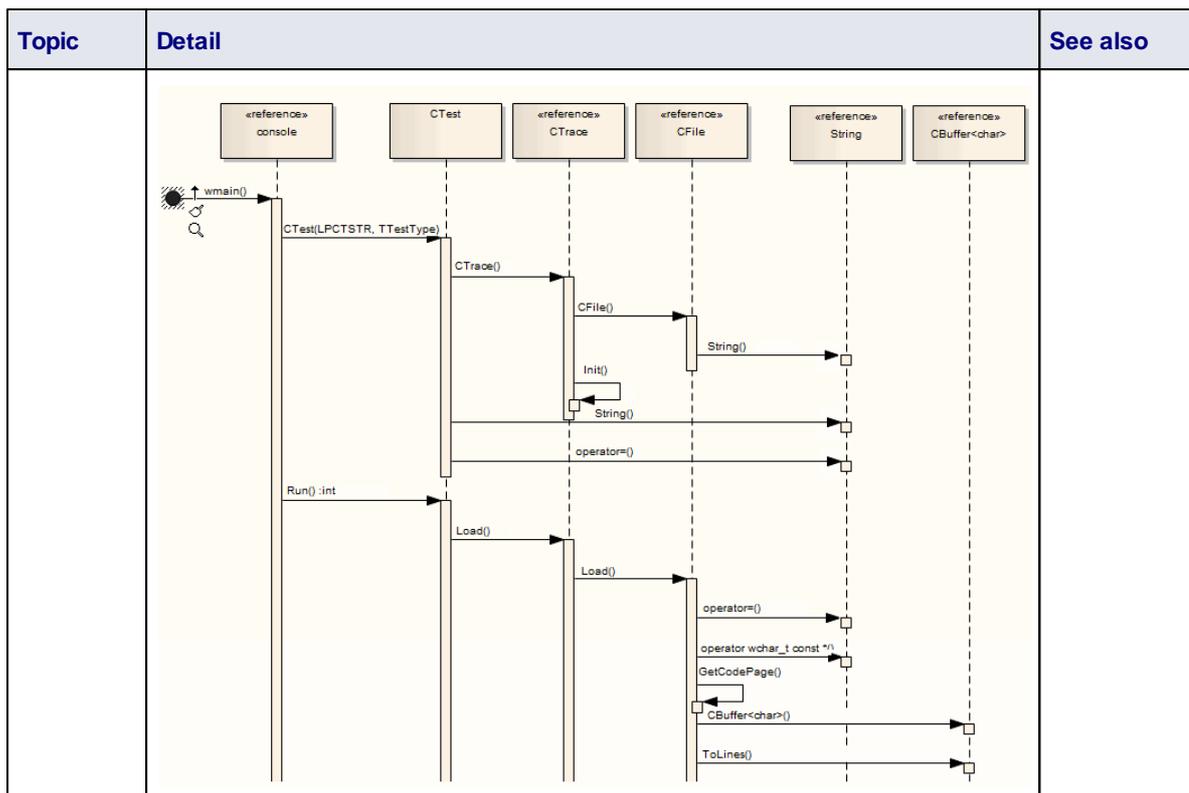
Recording markers are set in the source code editor. If you right-click on the breakpoint margin at the point to begin recording, a context menu displays that enables you to select the type of a marker.

Use To:

- Set Breakpoints and End points for recording
- Set Markers for a Single Method
- Set Stack Trace Markers

Topics:

Topic	Detail	See also
<p>Usage</p>	<p>Select the Add Start Recording Marker option, then right-click on the breakpoint margin at the point to stop recording and select the Add End Recording Marker context menu option; the markers are shown below:</p>  <pre> 17 private int m_delivery; 18 19 public ClassLib () { 20 21 } 22 23 /** 24 * 25 * @exception Throwable 26 */ 27 public void finalize () </pre> <p>When the debugger is executed it continues to run the thread, recording a stack history until either the End Recording marker is encountered or the thread terminates, unlike normal breakpoints where the debugger halts and displays the line of code</p> <p>In order to limit the amount of information being collected, you can control the stack depth being recorded.</p>	<p>Breakpoints and Markers <small>1646</small></p> <p>Analyzer Script Editor <small>1394</small></p> <p>Control Stack Depth <small>1641</small></p>
<p>Stack Auto-Capture Marker</p>	 <pre> 76 /* End - EA generated code for Parts and Ports */ 77 /* Begin - EA generated code for Activities and I 78 public void ClassLib_ActivityGraphWithActionPin () 79 { </pre> <p>(Native Code only) Stack markers enable you to capture any unique stack traces that occur at a point in an application; they provide a quick and useful picture of where a point in an application is being called from</p> <p>To insert a marker at the required point in code, right-click on the line and select the Add Stack Auto Capture Marker context menu option</p> <p>Each time the debugger encounters the marker it performs a stack trace; if the stack trace is not in the recording history, it is copied, and the application continues running</p>	

**Learn More:**

- [Recording Activity for a Single Method](#) ¹⁶⁴⁵
- [Recording Activity for a Class](#) ¹⁶⁴⁶

14.1.3.1.1 Set Record Markers

Markers are set in the code editor. Like breakpoints, they are placed on a line of code. When that line of code executes, the Execution Analyzer evaluates starts or stops recording code execution.

Step	Action	See also
1	Open the source code to debug in the integrated source code editor;	
2	Find the appropriate code line and right click in the left margin to bring up the Analyzer context menu. Select one of the following: <ul style="list-style-type: none"> • Add Start Recording Marker • Add End Recording Marker • Add Stack Auto Capture Recording Marker 	

Step	Action	See also
	<ul style="list-style-type: none">  Add Breakpoint  Add Start Recording Marker  Add End Recording Marker  Add Stack Auto Capture Marker  Add Method Auto Record Marker  Add Tracepoint Marker  Help 	
3	Where a Start Recording Marker has been set - you need to add an End Recording Marker.	

14.1.3.2 Recording Activity for a Single Method

A **Method Auto Record** marker enables you to record activity for a particular function during a debug session. The debugger records any function calls executed after the marker point, and always stops recording when this function exits. The function marker combines a **Start Recording** marker and an **End Recording** marker in one.

Example:

```

185 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
186 // CRecurrenceDlg message handlers
187
188 BOOL CRecurrenceDlg::OnInitDialog()
189 {
190     CBCGPDIALOG::OnInitDialog();
191
192     UINT nMask =
193         CBCGPDateTimeCtrl::DTM_SPIN      |
194         CBCGPDateTimeCtrl::DTM_DATE     |
195         CBCGPDateTimeCtrl::DTM_TIME     |
196         CBCGPDateTimeCtrl::DTM_CHECKBOX |
197         CBCGPDateTimeCtrl::DTM_DROPCALENDAR |
198         CBCGPDateTimeCtrl::DTM_CHECKED;
199
200     UINT nFlags = CBCGPDateTimeCtrl::DTM_CHECKED | CBCGPDateTimeCtrl::DI
201     //-----
202     // Setup date fields:

```

Learn More:

- [Marker Types](#) ¹⁶⁴²

14.1.3.3 Recording Activity for a Class

In addition to setting breakpoints and markers in the code editor, you can record all the operations of a Class or a subset using the Class Markup Selection dialog to create a *marker set*. Marker sets are available to all users of the model.

Access: [Project Browser Class Context Menu](#) | [Execution Analyzer](#) | [Markup Class For Recording](#)

Topics:

Topic	Detail	See also
Usage	<p>Type a name for the marker set, select the check box against each operation to record and, in the Marker Type field, click on the drop-down arrow and select the marker type</p> <p>The marker type specifies the action to take when the process encounters that marker:</p> <ul style="list-style-type: none"> • Record function • Record stack trace • Break execution <p>You can also specify a recording depth that limits the recording which, if uncontrolled, can ultimately produce Sequence diagrams that are too complicated to read; when you specify a depth, the Debugger does not record beyond this depth</p> <p>If necessary, you can override the recording depth from the Breakpoint & Markers window toolbar</p> <p>The depth is relative to the stack depth where the Debugger first encountered the recording marker; so, if the stack depth is 7 when recording begins, and the limit depth is set to 3, the Debugger does not record beyond a stack depth of 10</p> <p>Click on the OK button to store the marker set under the name you have specified; the set can then be loaded either before or during a session, again from the Breakpoint & Markers window toolbar</p>	Breakpoint & Markers ^[1646]

14.1.3.4 The Breakpoints & Markers Window

The Breakpoints & Markers window enables you to apply control to the Execution Analysis process.

Access: [Analyzer](#) | [Breakpoints & Markers](#) > [Breakpoints & Markers](#)

Topics:

Topic	Detail	See also
Usage	<p>Use the Breakpoints & Markers window to enable, disable and delete markers, and manage them as sets</p> <p>You can organize how the markers are displayed, either in list view or grouped by file or Class</p>	Activate and Disable Markers ^[1647] Working with Marker Sets ^[1648] Recording Activity for

Topic	Detail	See also
		a Class ^[1646]

Related Topics:

[Breakpoint and Marker Management](#) ^[1436]

14.1.3.4.1 Activate and Disable Markers

This topic describes how to delete, enable and disable breakpoints and markers.

Topics:

Topic	Detail	See also
Delete a breakpoint or marker	To delete a specific breakpoint: <ul style="list-style-type: none"> • If the breakpoint is enabled, click on the red breakpoint circle in the left margin of the Source Code Editor, or • Right-click on the breakpoint or marker in the Source Code Editor, the <i>Breakpoints</i> folder or the Breakpoints & Markers window and select the Delete context menu option, or • Select the breakpoint in the Breakpoints & Markers tab and press (Delete) 	Setting Code Breakpoints ^[1438] Breakpoint and Marker Management ^[1436]
Delete all breakpoints	Click on the Delete all breakpoints button on the Breakpoints & Markers window toolbar ()	
Convert breakpoint to Start Recording marker or End Recording marker	In the <i>Breakpoints</i> folder or the Breakpoints & Markers window, right-click on the breakpoint and select the context menu option to convert it to either a Start Recording marker or an End Recording marker	
Disable a breakpoint	Deselect the checkbox against the breakpoint or marker, on the Breakpoints & Markers window The breakpoint is then shown as an empty grey circle	
Enable a breakpoint or marker	Select the checkbox against the breakpoint or marker, on the Breakpoints & Markers window	
Disable all breakpoints	Click on the Disable all breakpoints button in the Breakpoints & Markers window toolbar ()	
Enable all breakpoints	Click on the Enable all breakpoints button in the Breakpoints & Markers window toolbar ()	

Learn More:

- [Marker Types](#) ^[1642]

14.1.3.4.2 Working with Marker Sets

Marker sets enable you to group markers into collections.

Topics:

Topic	Detail	See also
Usage	<p>A marker set can be used to record a specific Use Case, which might involve the operations of various Classes</p> <p>Once a set is created it is saved with the model; any user using the model has access to that set</p> <p>Sets are normally loaded prior to the point at which an action is to be captured; for example, to record a sequence involving a particular dialog, you might set markers for the areas to record, saving the markers as a set</p> <p>When you begin debugging, prior to invoking the dialog you would load the set; once you bring up the dialog in the application, the operations you have marked are recorded</p> <p>Review the recording history and create a Sequence diagram</p>	

14.1.4 Control the Recording Session

The Record & Analyze window enables you to control a recording session. The control has a toolbar and history window that displays the recording history as it is captured. Each entry in this window represents a call sequence made up of one or more function calls.

Access: **Analyzer | Record | Recorder**

You must also open the Execution Analyzer window (**Analyzer | Execution Analyzer**), which lists all the scripts in the model; make sure the active script is appropriate for the recording to make.

Learn More:

- [Recorder Toolbar](#)^[1648]
- [Start Recording](#)^[1650]
- [State Transition Recorder](#)^[1650]

14.1.4.1 Recorder Toolbar

The Record & Analyzer Toolbar is described below.

Reference:

Icon	Description	See also
	<p>The Recording icon is enabled when the active Analyzer Script is configured for debugging</p> <p>Use this icon to launch and record the application described in the script</p>	Start Recording ^[1650]

Icon	Description	See also
	You can optionally select a record marker set and / or a State machine to use during the recording	Recording Activity for a Class ^[1646]
	<p>The Recording (attach to process) icon is enabled even if no Analyzer Script exists</p> <p>When you use this icon, a prompt displays to select a process to record and a debugging platform to use</p> <p>You can also optionally select a record marker set and / or a State machine to use during the recording</p>	Start Recording ^[1650] Recording Activity for a Class ^[1646]
	<p>Use the Manual Record icon to perform ad-hoc recording of the current thread during a debug session</p> <p>Used in conjunction with the 'step' buttons of the debugger; each function that is called due to a step command is logged to the history window</p> <p>This icon is only enabled if no recording is taking place and you are currently at a breakpoint (that is, debugging)</p>	
	<p>Use the Auto Record Selected Thread icon to perform ad-hoc auto-recording during a debug session</p> <p>When you select this icon, the Analyzer begins recording and does not stop until either the program ends, you stop the debugger or you click on the Stop icon</p> <p>This icon is only enabled if no recording is taking place and you are currently at a breakpoint (that is, debugging)</p>	
	<p>Use the Step Through icon to step into a function, record the function call in the History window, and step back out</p> <p>Enabled for manual recording only</p>	Step Through Function Calls ^[1449]
	Use the Stop Recording icon to stop recording the execution trace	
	Use this field to define the marker set's recording stack depth - the number of frames from the point recording began	Marker Types ^[1642]
	<p>Select the Show/Hide lines during recording icon to show live line-by-line execution in the code editors as the program runs</p> <p>This icon is disabled if marker sets are in use</p>	
	Use this icon to select an XML file from which to load a previously-saved recording history	
	Use the Save Recording History to file icon to save the recording history to an XML file	Saving Recording ^[1653]
	Use the Generate Sequence Diagram from recording icon to generate a Sequence/State diagram from the Execution Analyzer trace	Generate a Diagram ^[1652]
	Use the Testpoints icon to generate a TestPoints diagram from the Execution Analyzer trace, including Classes, Test Cuts and a Test Set that can be used with the Testpoint facility	Testpoint Management ^[1672] Generate a Diagram ^[1652]

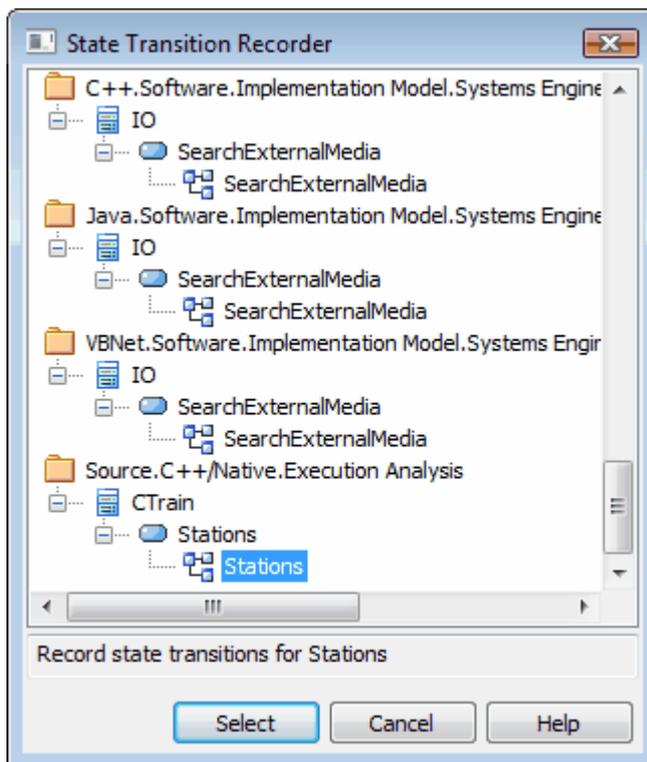
14.1.4.2 Start Recording

When you begin recording the Record dialog displays, on which you can optionally select a marker set to use and choose State Transitions to record during the session. The Execution Analyzer captures instances of the Class that the State Machine refers to and calculates the State whenever a function in the recorded stack exits.

Button	Detail
Using record set	Recording markers determine what is recorded If you have a recording set to use you can select it here
Using Statechart / Diagram	The recording can also capture changes in State transitions If you have a State Machine to use you can select it by clicking on this button

14.1.4.3 State Transition Recorder

The State Transition dialog presents a list of State Machines for the entire model. The Execution Analyzer can capture State Transitions designed on the Statechart while it is recording. When the Sequence Diagram is generated the diagram depicts not only the sequence but changes in State at the various points in the sequence. Each Class instance participating in the detection process is displayed with its own lifeline.



14.1.5 Generating Sequence Diagrams

Topics:

Topic	Detail	See also
Usage	<p>Once you have captured activity and are about to generate the diagram:</p> <ul style="list-style-type: none"> Select a package in the Project Browser where the Sequence diagram is to be stored Use the Generate Sequence Diagram from recording button on the Record & Analyze window toolbar to generate the diagram 	Recorder Toolbar <small>1648</small>

14.1.5.1 The Recording History

All information recorded as a result of the application encountering recording markers set by the user is held in the Record & Analyze window.

Topics:

Topic	Detail	See also
Columns	<p>The columns in this Record & Analyze window are as follows:</p> <ul style="list-style-type: none"> Sequence - The unique sequence number Threads - The operating system thread ID Delta - The elapsed thread CPU time since the start of the sequence Method - There are two Method columns: the first shows the caller for a call or for a current frame if a return; the second shows the function called or function returning Direction - <i>Stack Frame Movement</i>, can be <i>Call</i>, <i>Return</i>, <i>State</i>, <i>Breakpoint</i> or <i>Escape</i> (<i>Escape</i> is used internally when producing a Sequence diagram, to mark the end of an iteration) Depth - The stack depth at the time of a call; used in the generation of Sequence diagrams State - The state between sequences Source - There are two Source columns: the first shows the source filename and line number of the caller for a call, or for a current frame if a return; the second shows the source filename and line number of the function called or function returning Instance - There are two Instance columns; these columns only have values when the Sequence diagram produced contains State transitions. The values consist of two items separated by a comma - the first item is a unique number for the instance of the Class that was captured, and the second is the actual instance of the Class <p>For example: supposing a Class <i>CName</i> has an internal value of 4567 and the program created two instances of that Class; the values might be:</p> <ul style="list-style-type: none"> 4567,1 	

Topic	Detail	See also
	<ul style="list-style-type: none"> 4567,2 <p>The first entry shows the first instance of the Class and the second entry shows the second instance</p>	
Toolbar	The Record & Analyze window toolbar provides a range of facilities for controlling the recording of the execution of an Analyzer script	Recorder Toolbar ^[1648]

Notes:

- The checkbox against each number is used to control whether or not this call should be used to create a Sequence diagram from this history
- In addition to enabling or disabling the call using the checkbox, you can use context menu options to enable or disable an entire call, all calls to a given method, or all calls to a given Class

14.1.5.2 Generate a Diagram**Topics:**

Topic	Detail	See also
Usage	<p>To generate a Sequence diagram for:</p> <ul style="list-style-type: none"> All history, click on the toolbar Create Sequence Diagram icon () A single sequence, select it and then click the toolbar Create Sequence Diagram icon () 	The Recording History ^[1651]

14.1.5.3 Diagram Features

When you generate a Sequence diagram, it includes the following features:

Feature	Detail	See also
References	<p>When the Visual Execution Analyzer cannot match a function call to an operation within the model, it still creates the sequence but also creates a reference for any Class that it cannot locate</p> <p>It does this for all languages</p>	
Fragments	<p>Fragments displayed in the Sequence diagram represent loops or iterations of a section(s) of code</p> <p>The Visual Execution Analyzer attempts to match function scope with method calls to as accurately as possible represent the execution visually</p>	
States	<p>If a State Machine has been used during the recording process, any transitions in State are presented after the method call that caused the transition to occur</p> <p>States are calculated on the return of every method to its caller</p>	

14.1.5.4 Saving Recording

This topic explains how to save a recorded sequence to an XML file, and retrieve it.

Action	Detail	See also
To Save	To save a recorded sequence to an XML file, click on the sequence and on the toolbar Save icon ()	Recorder Toolbar ^[1648]
To Access	To access an existing sequence XML file, either: <ul style="list-style-type: none"> • Click on the toolbar Open icon (), or • Right-click on a blank area of the screen and click on the Load Sequence From File context menu option The Windows Open dialog displays, from which you select the file to open	Recorder Toolbar ^[1648]

14.1.6 Add State Transitions

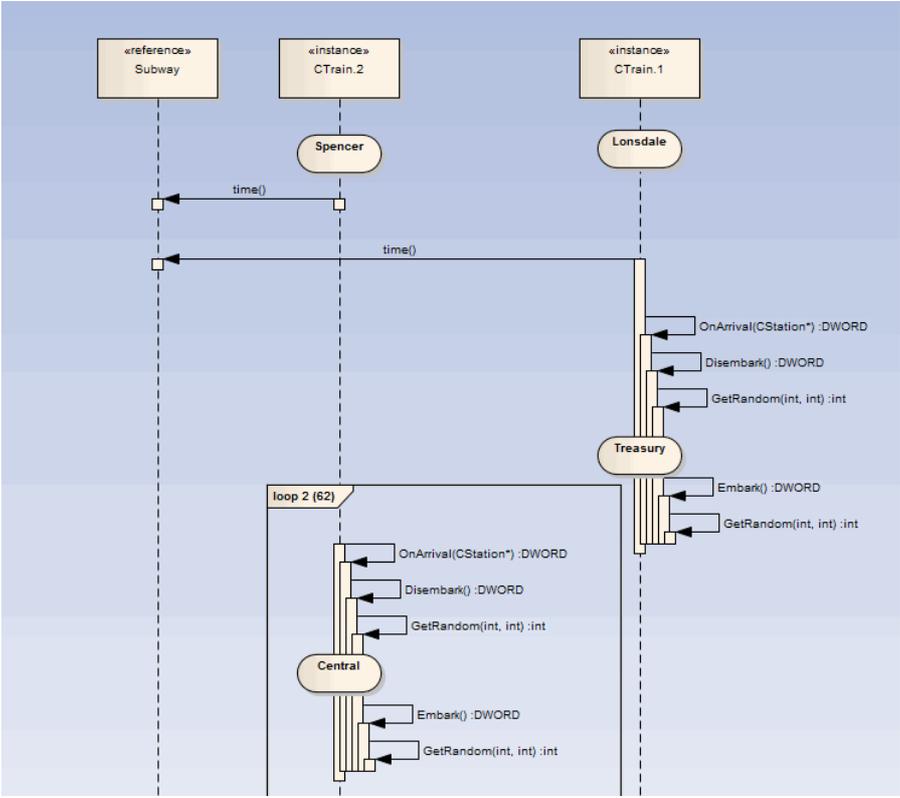
This section describes how you can generate Sequence diagrams that show transitions in state as a program executes.

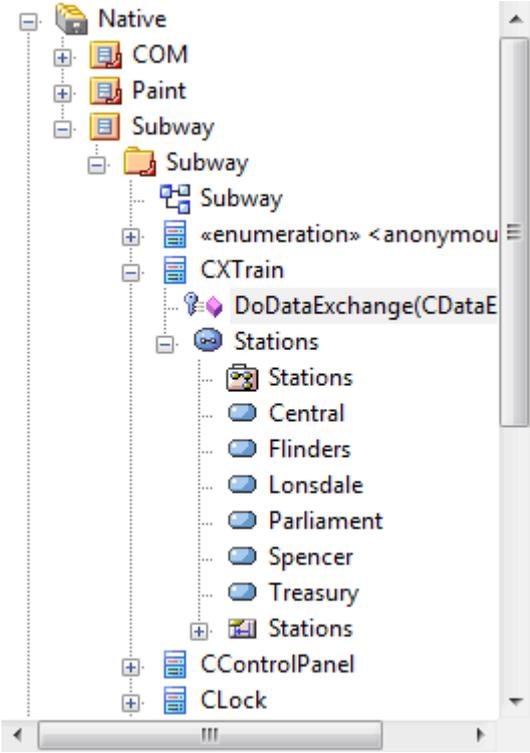
Topic	Link
Setup for capturing changes	Setup for Capturing State Changes ^[1653]
State change during the execution of an application	The State Machine ^[1655]
Recording and Mapping state changes	Recording and Mapping State Changes ^[1657]

14.1.6.1 Setup for Capturing State Changes

You can generate Sequence diagrams that show transitions in state as a program executes. The illustration below shows a project that has, in its State Machine, a number of States that correspond to stations in the Melbourne underground railway system.

Topics:

Topic	Detail	See also
<p>State Transitions</p>	 <p>Showing State transitions on your debug-generated Sequence diagrams is optional; you set an option in the package script associated with the Class for which you intend to record States</p> <p>You then create a State Machine under the Class, and on the State Machine you create the State elements that correspond to any states to be captured for your Class</p> <p>The debugger evaluates your States by checking constraints on the States you create; the States on this diagram are then used by the debugger and State transitions are incorporated into the diagram</p>	
<p>Example</p>		

Topic	Detail	See also
	 <p>This figure shows the Class <i>CXTrain</i> with a State Machine called <i>Stations</i>; it has a child diagram also called <i>Stations</i>, on which the States {<i>Central</i>, <i>Flinders</i>, <i>Lonsdale</i>...} are placed</p>	

Notes:

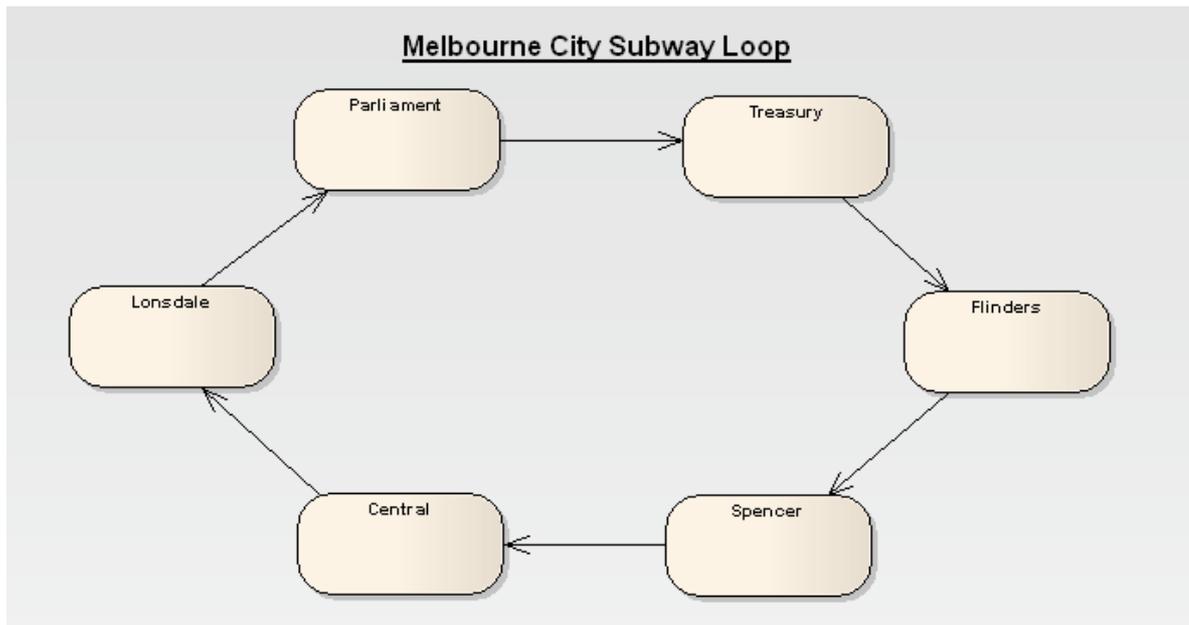
- If you do not have a package script for the Class or package you must create one; Sequence diagrams can only be generated for a package that has been configured for debug

14.1.6.2 The State Machine

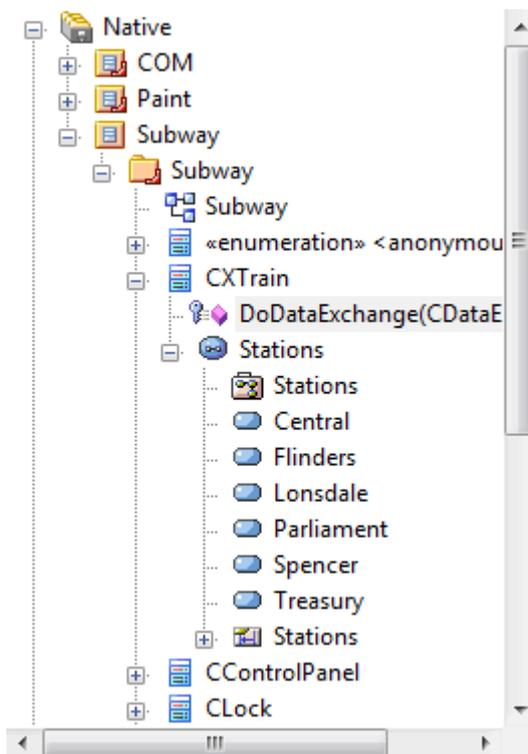
A State Transition diagram can be used to illustrate how States change during the execution of an application. The Visual Execution Analyzer can build a State Machine to model all the valid system states and explicitly describe the transitions between each state.

The diagram below is a State Machine that shows the different States within the Melbourne Underground Loop subway system. A train traveling on the subway network can be stopped at any of the stations represented on the State Machine.

Example:



This State Machine diagram is a child of the *CXTrain* Class.



14.1.6.3 Recording and Mapping State Changes

Topics:

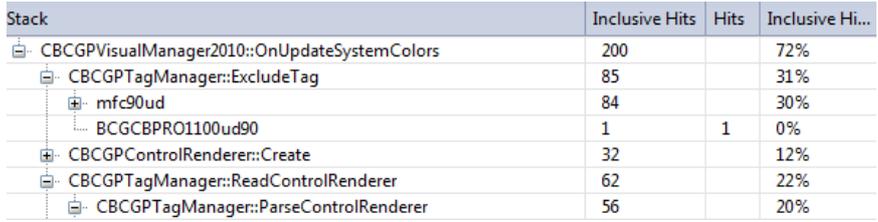
Topic	Detail	See also									
<p>Example</p>	<p>The State Properties dialog shown below is for the State called <i>Parliament</i>; the Constraints tab is open to show how the State is linked to the Class <i>CXTrain</i></p> <p>A State can be defined by a single constraint or by many; in the example below the State <i>Parliament</i> has two constraints:</p> <div data-bbox="483 683 1289 862" style="border: 1px solid gray; padding: 5px;"> <p>Defined Constraints   New Save Delete</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 70%;">Constraint</th> <th style="width: 15%;">Type</th> <th style="width: 15%;">Status</th> </tr> </thead> <tbody> <tr> <td>Location=0</td> <td>Invariant</td> <td>Approved</td> </tr> <tr> <td>Departing.Name=Parliament</td> <td>Invariant</td> <td>Approved</td> </tr> </tbody> </table> </div> <p>The values of constraints can only be compared for <i>elemental</i>, <i>enum</i> and <i>string</i> types</p> <p>The <i>CXTrain</i> Class has a member called <i>Location</i> of type <i>int</i>, and a member called <i>Departing.Name</i> of type <i>CString</i>; what this constraint means is that this State is evaluated to true when:</p> <ul style="list-style-type: none"> • an instance of the <i>CXTrain</i> Class exists and • its member variable <i>Location</i> has the value 0 and • the member variable <i>Departing.Name</i> has the value Parliament 	Constraint	Type	Status	Location=0	Invariant	Approved	Departing.Name=Parliament	Invariant	Approved	
Constraint	Type	Status									
Location=0	Invariant	Approved									
Departing.Name=Parliament	Invariant	Approved									
<p>Operators in Constraints</p>	<p>There are two types of operators you can use on constraints to define a State:</p> <ul style="list-style-type: none"> • Logical operators AND and OR can be used to combine constraints • Equivalence operators {= and !=} can be used to define the conditions of a constraint <p>All the constraints for a State are subject to an AND operation unless otherwise specified; you can use the OR operation on them instead, so you could rewrite the constraints in the above example as:</p> <pre>Location=0 OR Location=1 AND Departing.Name!=Central</pre> <p>Below are some examples of using the equivalence operators:</p> <pre>Departing.Name!=Central AND Location!=1</pre>										

Notes:

- Quotes around strings are optional; the comparison for strings is always case-sensitive in determining the truth of a constraint

14.2 Profiling Native Applications

Topics:

Topic	Detail	See also																																
Visual Execution Profiler	<p>The Visual Execution Profiler enables you to quickly report on:</p> <ul style="list-style-type: none"> • The most frequently called functions in a running application • Tasks in an application that are taking more time than expected • Which functions are taking the most time in an application <p>The Profiler only works with MS Native Windows applications, but can be used under WINE (Linux and Mac) to debug standard Windows applications deployed in a WINE environment</p> <p>The Profiler can generate a report that shows how these functions are called in relation to the application; the following image shows how one function is taking most of the overall time recorded, and how that is broken up into some of the functions that it calls</p>  <table border="1"> <thead> <tr> <th>Stack</th> <th>Inclusive Hits</th> <th>Hits</th> <th>Inclusive Hi...</th> </tr> </thead> <tbody> <tr> <td>CBCGPVisualManager2010::OnUpdateSystemColors</td> <td>200</td> <td></td> <td>72%</td> </tr> <tr> <td> CBCGPTagManager::ExcludeTag</td> <td>85</td> <td></td> <td>31%</td> </tr> <tr> <td> mfc90ud</td> <td>84</td> <td></td> <td>30%</td> </tr> <tr> <td> BCGCBPRO1100ud90</td> <td>1</td> <td>1</td> <td>0%</td> </tr> <tr> <td> CBCGPControlRenderer::Create</td> <td>32</td> <td></td> <td>12%</td> </tr> <tr> <td> CBCGPTagManager::ReadControlRenderer</td> <td>62</td> <td></td> <td>22%</td> </tr> <tr> <td> CBCGPTagManager::ParseControlRenderer</td> <td>56</td> <td></td> <td>20%</td> </tr> </tbody> </table>	Stack	Inclusive Hits	Hits	Inclusive Hi...	CBCGPVisualManager2010::OnUpdateSystemColors	200		72%	CBCGPTagManager::ExcludeTag	85		31%	mfc90ud	84		30%	BCGCBPRO1100ud90	1	1	0%	CBCGPControlRenderer::Create	32		12%	CBCGPTagManager::ReadControlRenderer	62		22%	CBCGPTagManager::ParseControlRenderer	56		20%	
Stack	Inclusive Hits	Hits	Inclusive Hi...																															
CBCGPVisualManager2010::OnUpdateSystemColors	200		72%																															
CBCGPTagManager::ExcludeTag	85		31%																															
mfc90ud	84		30%																															
BCGCBPRO1100ud90	1	1	0%																															
CBCGPControlRenderer::Create	32		12%																															
CBCGPTagManager::ReadControlRenderer	62		22%																															
CBCGPTagManager::ParseControlRenderer	56		20%																															

Notes:

- The Profiler, or sampler, is available in the Enterprise Architect Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions

Learn More:

- [Profiler System Requirements](#) ^[1658]
- [Profiler Operation](#) ^[1661]

14.2.1 System Requirements

Topics:

Topic	Detail	See also
Prerequisites	<p>The Profiler window becomes available when a model is opened</p> <p>Options on the Profiler window toolbar enable you to attach to an existing process, or to launch a new application if a Package Script been specified</p>	Profiling Native Applications ^[1658] Getting Started ^[1659]
Supported	Enterprise Architect supports profiling on native Windows	

Topic	Detail	See also
Platforms	<p>applications (C, C++ and Visual Basic) compiled with the Microsoft™ native compiler, where an associated PDB file is available; select Microsoft Native from the list of debugging platforms in your package script</p> <p>The Profiler can sample both Debug and Release configurations of an application, providing the PDB for each executable exists and is up to date</p>	

14.2.2 Getting Started

Access: [Analyzer](#) | [Profile](#)

Topics:

Topic	Detail	See also
Abstract	<p>The Profiler operates by taking samples of a process at intervals of up to 250 milliseconds</p> <p>At these intervals the Profiler interrupts the process and collects stack information for all threads running at that time; this information is sent back to Enterprise Architect where it is collected sorted and stored</p> <p>You can pause and resume profiling at any time during the session, or clear any sample data collected and begin again</p> <p>If you stop the Profiler and the process is still running, you can quickly attach to it again</p>	

Toolbar Options:

Icon	Action	See Also
	(When an application is configured for the package) create the Profiler process, which launches the configured application	Managing Scripts <small>[1392]</small>
	Profile an application that is already running	
	When the application is running, pause and resume sample capture Pausing sampling enables the Report and Erase buttons	
	Stop the Profiler process; if any samples have been collected, the Report button is enabled	
	Generate a report on the current number of samples collected	Profiling Native Application <small>[1658]</small>
	Set the interval, in milliseconds, at which samples are taken of the target process; the range of possible values is 1 - 250	Setting Options <small>[1661]</small>

Icon	Action	See Also
	Set Profiler options, using a drop-down menu; the options are: <ul style="list-style-type: none"> • Load Report from Disk - load and display a previously-generated report from an XML disk file • Analyzer Scripts (Shift+F12) - display the Execution Analyzer window to enable creation or editing of scripts and debug configuration • Start Sampling Immediately - begin sample collection immediately upon either process start (main thread entry point executed) or attachment of process by Profiler • Capture Debug output - capture any appropriate debug output and redirect it to the Enterprise Architect Output window • Stop Process on Exit - select to terminate the target process when the Profiler is stopped 	Build ^[1415]
	Erase the collected data	
	Display the Help topic for this window	

14.2.3 Start & Stop the Profiler

Topics:

Topic	Detail	See also
Usage	<p>For most debugging operations it is necessary to have first configured a Package Script that typically defines the application to build, test and debug, and any sequence recording options</p> <p>It is possible to use the Profiler without doing any of this by using the Attach to Process button</p> <p>If the application to profile is the one defined in the current package, use the Launch button</p> 	
Buttons	 (When an application is configured for the package) create the Profiler process, which launches the configured application	Managing Scripts ^[1392]
	 Profile an application that is already running	
	 Stop the Profiler process	

14.2.4 Profiler Operation

Topics:

Topic	Detail	See also
Usage	<p>Enterprise Architect creates a Profiler process whenever you click on the Launch or Attach to Process button on the Profiler window toolbar; this process operates by collecting samples from the stacks of every thread in the target process</p> <div data-bbox="349 633 1195 817" data-label="Diagram"> <pre> classDiagram class EA class Profiler class TargetApplication TargetApplication --> Profiler : «flow» Profiler --> EA : «flow» </pre> </div> <p>The sampler process exits if you click on the Stop button, if the target application terminates, or if you close the current model</p> <p>You can turn sample collection on and off at any time during a session</p> <p>When sampling is turned on or resumed, the Profiler process becomes active and samples are collected from the target; resuming sampling collects completely new samples</p> <p>The Profiler process idles if sampling is turned off or paused during a session. The Report and Erase buttons then become enabled</p> <p>Click on the Report button to produce a call graph summary similar to that in the <i>Profile Native Applications</i> topic; this report can be saved to file</p> <p>Click on the Erase button to discard any samples currently collected for the target</p>	<p>Profile Native Applications </p>

14.2.5 Setting Options

Topics:

Topic	Icon	Detail	See also
Interval		Set the interval, in milliseconds, at which samples are taken of the target process; the range of possible values is 1 - 250	
Profile Options		<p>Set Profiler options, using a drop-down menu; the options are:</p> <ul style="list-style-type: none"> • Start Sampling Immediately - begin sample collection immediately upon either process start (main thread entry point executed) or attachment of process by Profiler • Capture Debug output - capture any appropriate debug output and redirect it to the Enterprise Architect Output window • Stop Process on Exit - select to terminate the target process when the Profiler is stopped 	

14.2.6 Save and Load Reports

Topics:

Topic	Detail	See also																																
Usage	<p>The Profiler Reports can be saved in either binary format or xml format.; save the report using the toolbar above the report (Stack) view</p> <table border="1"> <thead> <tr> <th>Stack</th> <th>Inclusive Hits</th> <th>Hits</th> <th>Inclusive Hi...</th> </tr> </thead> <tbody> <tr> <td>[-] CBCGPVisualManager2010::OnUpdateSystemColors</td> <td>200</td> <td></td> <td>72%</td> </tr> <tr> <td>[-] CBCGPAdapterManager::ExcludeTag</td> <td>85</td> <td></td> <td>31%</td> </tr> <tr> <td>[-] mfc90ud</td> <td>84</td> <td></td> <td>30%</td> </tr> <tr> <td>[-] BCGCBPRO1100ud90</td> <td>1</td> <td>1</td> <td>0%</td> </tr> <tr> <td>[-] CBCGPControlRenderer::Create</td> <td>32</td> <td></td> <td>12%</td> </tr> <tr> <td>[-] CBCGPAdapterManager::ReadControlRenderer</td> <td>62</td> <td></td> <td>22%</td> </tr> <tr> <td>[-] CBCGPAdapterManager::ParseControlRenderer</td> <td>56</td> <td></td> <td>20%</td> </tr> </tbody> </table> <p>To load a report, use the Profiler Toolbar Options button  and select the Load Report From Disk option</p>	Stack	Inclusive Hits	Hits	Inclusive Hi...	[-] CBCGPVisualManager2010::OnUpdateSystemColors	200		72%	[-] CBCGPAdapterManager::ExcludeTag	85		31%	[-] mfc90ud	84		30%	[-] BCGCBPRO1100ud90	1	1	0%	[-] CBCGPControlRenderer::Create	32		12%	[-] CBCGPAdapterManager::ReadControlRenderer	62		22%	[-] CBCGPAdapterManager::ParseControlRenderer	56		20%	
Stack	Inclusive Hits	Hits	Inclusive Hi...																															
[-] CBCGPVisualManager2010::OnUpdateSystemColors	200		72%																															
[-] CBCGPAdapterManager::ExcludeTag	85		31%																															
[-] mfc90ud	84		30%																															
[-] BCGCBPRO1100ud90	1	1	0%																															
[-] CBCGPControlRenderer::Create	32		12%																															
[-] CBCGPAdapterManager::ReadControlRenderer	62		22%																															
[-] CBCGPAdapterManager::ParseControlRenderer	56		20%																															

14.2.7 Save Report in Team Review

You can save any current report as a resource for a Category, Topic or Post in the Team Review. The report can then be shared and reviewed at any time as it is saved with the model.

Access: [Team Review Context Menu](#) | [Share Resource](#) | [Add Active Profiler Report](#)

Use to:

- Preserve a profiler report to compare against future runs
- Allow other people to investigate the profile

Learn More:

- [\(Team Review\) Context Menu](#)²¹⁸

14.3 Object Workbench

This section describes the Object Workbench, a tool in Enterprise Architect Debugging that enables you to create your own variables and invoke methods on them.

Topic	Link
Setup requirements for the workbench	Workbench Setup <small>[1663]</small>
Overview - your methods can be invoked to record Stack trace and produce Sequence diagrams	How it works <small>[1663]</small>
Description of workbench variables	Workbench Variables <small>[1665]</small>
How to create workbench variables	Create Workbench Variables <small>[1665]</small>
How to invoke methods	Invoke Methods <small>[1666]</small>

14.3.1 Workbench Setup

This topic describes the requirements for setting up the Object Workbench on Java and Microsoft .NET.

Java Workbench:

To use the Object Workbench you must have created an Analyzer Script for the package and configured the Debugger.

The Java workbench uses the Virtual Machine settings configured in the Analyzer Script Debug page to create the JVM.

Microsoft .NET Workbench:

You must have created an Analyzer Script for the package and configured the Debugger.

The .NET workbench requires an assembly, which is used to create the workbench items. You specify the path to the assembly on the Workbench page of the Analyzer Script.

Debug Platform: Microsoft Native

Microsoft .NET

The specified assembly is used to create the Workbench Objects.

Java

The workbench uses the same runtime options defined in the debugging section.

Native

Not supported.

The Workbench Page of the Analyzer Script

14.3.2 How it Works

The Workbench is a tool in Enterprise Architect Debugging, enabling you to create your own variables and invoke methods on them. Stack trace can be recorded and Sequence diagrams produced from the invocation of such methods. It provides a quick and simple way to debug your code.

Topics:

Topic	Detail	See also
Platforms Supported	<p>The Workbench supports the following workbench platforms:</p> <ul style="list-style-type: none"> • Microsoft .NET (version 2.0 or later) • Java (JDK 1.4 or later) <p>The Workbench does not currently support the creation of Class instances written in native C++, C or VB</p>	
Mode	<p>The Workbench operates in two modes:</p> <ul style="list-style-type: none"> • Idle mode - When the Workbench is in idle mode, instances can be created and viewed and their members inspected • Active mode - When methods are invoked on an instance, the Workbench enters Active mode and the displayed variables change if the debugger encounters any breakpoints; if no breakpoints are set, then the variables do not change and the Workbench immediately returns to Idle mode 	
Logging	<p>The results of creating variables and the results of calls on their methods are displayed in the Debug Output window</p>	<p>Show Output From Debugger [1446]</p>

Learn More:

- [Create Workbench Variables](#) [1665]
- [Invoke Methods](#) [1666]

14.3.3 Workbench Variables

You can create (and delete) workbench variables from any Class in your model. When you do so, you are asked to name the variable. It then displays in the Workbench window, showing the variable in a hierarchy and displaying its type and value and those of any members.

Topics:

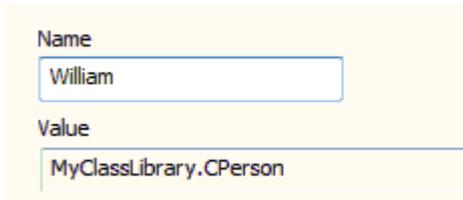
Topic	Detail	See also
Workbench Requirements	<ul style="list-style-type: none"> .NET framework version 2 is required to workbench any .NET model The package from which the variable is created must have a debugger configured 	Setup For Debugging ^[1417]
Constraints (.NET)	<ul style="list-style-type: none"> Members defined as <i>struct</i> in managed code are not supported Classes defined as <i>internal</i> are not supported 	
Delete Workbench Variables	<p>You can delete variables using the Delete shortcut menu on any instance on the Workbench</p> <p>If all instances are deleted, the debugger is shut down and the Workbench window is closed</p>	

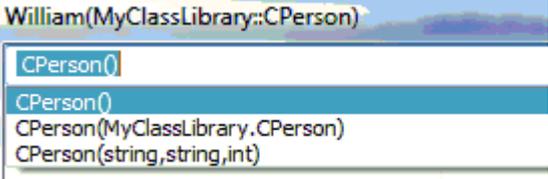
14.3.4 Create Workbench Variables

This topic explains how to create a workbench instance for a Class element.

Access: **Project Browser Class context menu | Execution Analyzer | Create Workbench Instance of Class (Ctrl+Shift+J)**

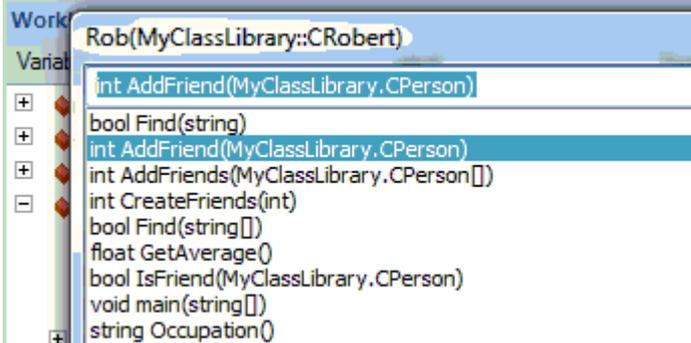
Topics:

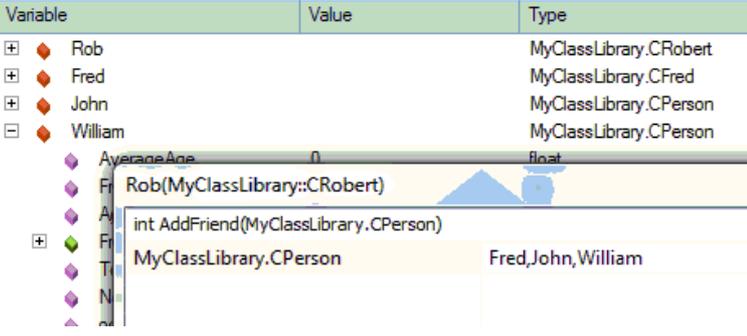
Topic	Detail	See also
Naming the Workbench	<p>Select the Create Workbench Instance of Class option</p> <p>The Workbench dialog displays</p>  <p>Type in a name for the variable; each instance name must be unique for the workbench</p>	
Choosing a Constructor	<p>After naming the variable, select the constructor to use</p> <p>If you do not define a constructor, or define a single constructor taking no arguments, the default constructor or the defined constructor is automatically invoked</p> <p>Otherwise the following dialog displays; select the constructor from the drop-</p>	

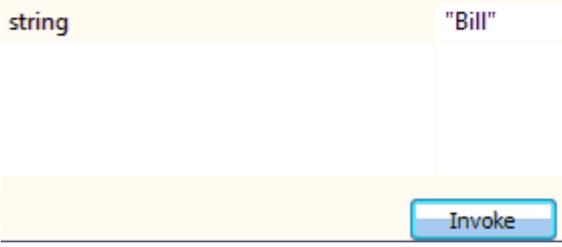
Topic	Detail	See also
	<p>down list and fill in any parameters required</p> 	

14.3.5 Invoke Methods

Topics:

Topic	Detail	See also
Access	On the Workbench window, right-click on the instance on which to execute a method, and select the Invoke context menu option	
Choose Method	<p>A list of methods for the type are presented in a dialog; select a method from the list and click on the Invoke button</p> <p>Note that all methods listed are public; private methods are not available</p> 	
Supply Arguments	<p>In the example, the instance or variable <i>Rob</i> has been created, of type <i>MyClassLibrary.CRobert</i>, and the method <i>AddFriends</i> invoked that takes an array of <i>CPerson</i> objects as its only argument</p> <p>You now supply to it three other Workbench instances: <i>Fred</i>, <i>John</i> and <i>William</i></p>	

Topic	Detail	See also						
								
Arguments	<p>In the dialog above, type any parameters required by the constructor:</p> <ul style="list-style-type: none"> • Literals as arguments <ul style="list-style-type: none"> • Text: abc or "abc" or "a b c" • Numbers: 1 or 1.5 • Objects as arguments <p>If an argument is not a literal then you can supply it in the list only if you have already created an instance of that type in the workbench; you do this by typing the name of the instance as the argument</p> <p>The debugger checks any name entered in an argument against its list of workbench instances, and substitutes that instance in the actual call to the method</p> • Strings as arguments <p>Surrounding strings with quotes is unnecessary as anything you type for a string argument becomes the value of the string; for example, the only time you should surround strings with quotes is in supplying elements of a string array, or where the string is equal to the name of an existing workbench instance</p> <pre> "A b c" "a b \$ % 6 4" A b c d As 5 7) 2 === 4 </pre> • Arrays as arguments <p>Enter the elements that compose the array, separated by commas</p> <table border="0" data-bbox="555 1803 1177 1899"> <tr> <td>Type</td> <td>Arguments</td> </tr> <tr> <td>String()</td> <td>one,two,three,"a book","a bigger book"</td> </tr> <tr> <td>CPerson()</td> <td>Tom,Dick,Harry</td> </tr> </table> <p>If you enter text that matches the name of an existing instance, surround it in quotes to avoid the debugger passing the instance rather than a string</p> 	Type	Arguments	String()	one,two,three,"a book","a bigger book"	CPerson()	Tom,Dick,Harry	
Type	Arguments							
String()	one,two,three,"a book","a bigger book"							
CPerson()	Tom,Dick,Harry							

Topic	Detail	See also
	 <p>The screenshot shows a code editor with the method signature <code>void SetName(string)</code>. Below it, a parameter <code>string</code> is highlighted in yellow, and the value <code>"Bill"</code> is entered in a text box. At the bottom, an <code>Invoke</code> button is visible.</p>	
Invoke	Having chosen the constructor and supplied any arguments, click on the Invoke button to create the variable Output confirming this action is displayed in the Output tab	Show Output From Debugger <small>1446</small>

14.4 Unit Testing

Enterprise Architect supports integration with unit testing tools in order to make it easier to develop good quality software.

In sequence:

- Enterprise Architect helps you to create test Classes with the **JUnit** and **NUnit** transformations
- You set up and run a **test script** against any package
- All test results are automatically recorded inside Enterprise Architect

Learn More:

- [JUnit Transformation](#)^[1321]
- [NUnit Transformation](#)^[1323]
- [Setup Unit Testing](#)^[1669]
- [Add Testing Command](#)^[1669]
- [Run Unit Tests](#)^[1670]
- [Record Test Results](#)^[1670]

14.4.1 Set Up Unit Testing

Topics:

Topic	Detail	See also
Define Test Cases	<p>Firstly the appropriate tests must be defined; Enterprise Architect is able to help with this</p> <p>By using the JUnit or NUnit transformations and code generation you can create test method stubs for all of the public methods in each of your Classes</p> <p>The following is an <i>NUnit</i> example in C# that is followed through the rest of this topic, although it could also be any other .NET language, or Java and JUnit</p> <pre> (Test Fixture) public class CalculatorTest { (Test) public void testAdd(){ Assert.AreEqual (1+1, 2); } (Test) public void testDivide(){ Assert.AreEqual (2/ 2, 1); } (Test) public void testMultiply(){ Assert.AreEqual (1* 1, 1); } (Test) public void testSubtract(){ Assert.AreEqual (1- 1, 1); } } </pre>	<p>Adding Testing Command^[1451]</p> <p>JUnit Transformation^[1321]</p> <p>NUnit Transformation^[1323]</p> <p>Generate Source Code^[1490]</p>

Topic	Detail	See also
	This code can be reverse engineered into Enterprise Architect so that Enterprise Architect can record all test results against this Class	
Set up the Test Scripts	<p>Once the unit tests are set up, you can then set up the Build and Test scripts against a package, to run the tests</p> <p>The sample above can be called by setting up an Analyzer script with a command similar to the following:</p> <pre>" C : \ P r o g r a m F i l e s \ N U n i t \ b i n \ n u n i t - c o n s o l e . e x e " p i n \ d e b u g \ C a l c u l a t o r . e x e "</pre> <p>If Enterprise Architect is to handle unit testing, it is important that you select the Capture Output checkbox and select the appropriate Output Parser for the testing; without doing this you cannot see the program output and therefore you cannot open the source at the appropriate location</p>	Add Commands <small>[1415]</small>

14.4.2 Run Unit Tests

Access: [Analyzer](#) | [Test](#) | [Run Test Script](#)

Topics:

Topic	Detail	See also
Run a Unit Test	<p>Run the NUnit test script you set up previously</p> <p>The results of NUnit are displayed in the Output window, identifying which tests have run and which of these have failed</p> <p>The results also show which method failed and the file and line number the failure occurred at</p> <p>Double-click on an error message; Enterprise Architect opens the editor to that line of code, enabling you to quickly find and fix the error</p> <p>Enterprise Architect also records the run status of each test</p>	Record Test Results <small>[1670]</small>

14.4.3 Record Test Results

Enterprise Architect is able to automatically record all results from tests through a testing script in Enterprise Architect.

Topics:

Topic	Detail	See also
Process	<p>In order to use this feature, you must reverse engineer the test Class into the package containing your test script</p> <p>Once your model contains your test Class, on the next run of the test script Enterprise Architect adds Test Cases to the Class for each</p>	Set Up Unit Testing <small>[1669]</small> Importing Source Code <small>[1508]</small>

Topic	Detail	See also
	<p>test method found; on this and all subsequent test runs all Test Cases are updated with the current run time and whether they passed or failed, as shown below:</p> <div data-bbox="475 456 743 824" style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"><p style="text-align: center;">Calculator Test</p><ul style="list-style-type: none">+ testAdd() : void+ testDivide() : void+ testMultiply() : void+ testSubtract() : void<p style="text-align: center;">test scripts</p><ul style="list-style-type: none">Unit : (Pass) testAddUnit : (Pass) testDivideUnit : (Pass) testMultiplyUnit : (Fail) testSubtract</div> <p>The error description for each failed test is added to any existing results for that Test Case, along with the current date and time</p> <p>Over time this provides a log of all test runs where each Test Case has failed, which can then be included in generated documentation, resembling the following:</p> <pre>Failed at 05- Jul - 2006 1: 02: 08 PM expected: <0> but was: <1> Failed at 28- Jun- 2006 8: 45: 36 AM expected: <0> but was: <2></pre>	<p>Run Unit Tests [1670]</p>

14.5 Testpoint Management

Testpoint Management provides the facility to test and pass or fail application tasks, viewing test results in real time as the program executes and results are saved.

Tests	Id	Invariant	Pre	Post	Author	Status	Evals	Passes	Fails
CTrain									
Invariant					smeagher	✓	3302	3301	1
Passengers > 0						✓			
void Create()									
CTrain(CNetwork*, TTr									
~CTrain()									
DWORD Disembark(int)		✓			smeagher	✓	6	6	
Pre-Condition						✓			
PeopleOFF > 0						✓			
DWORD Embark(int)		✓			smeagher	✓	6	6	
Pre-Condition						✓			
PeopleON > 0						✓			
DWORD Execute(CTrair									
void Exit()									

The image shows a Testpoint run in progress. The example was produced using the Example Model that ships with Enterprise Architect

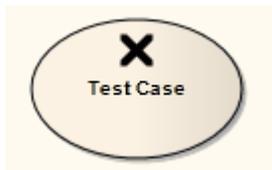
Topics:

Topic	Detail	See also
Overview	<p>You define tests as Class invariants, and as pre/post conditions on operations</p> <p>Testpoint design and management is performed on a Test Domain diagram, on which you define constraints and group them into sets, which in turn can form suites</p> <p>The Testpoint interface is contextual, and when an object (Class, Test Cut, Test Set, Test Suite) is selected, the constraints for that object are displayed</p> <p>In the Testpoint system, the rules that define task behaviors are defined within the model and do not form part of any code base, which has a number of advantages - changing a test condition in other systems can require that the application be re-built; in Enterprise Architect you could stop the run, change a constraint and start the run again without re-building the project</p>	
Test Domain Diagram	<p>A Test Domain diagram is a particular diagram designed for use with the Testpoint facility</p> <p>A Test Domain diagram provides specific elements that can aid in the logical composition of the Tests into Test sets and suites</p> <p>You can perform a Testpoint run on an entire Suite, a single Set or</p>	<p>Test Domain Group^[567]</p> <p>Test Cut^[1675]</p> <p>Test Set^[1675]</p>

Topic	Detail	See also
	an individual Class	Test Suite ^[1676]
Testpoint Class	<p>Test conditions are always defined on a Class; invariants are created for the Class itself, and pre- and post-conditions are defined on its operations</p> <p>These conditions can be aggregated later into sets, but the Classes are the building blocks upon which any sets you create are built</p>	
Constraints	<p>The Testpoint facility is based on a programming-by-contract model</p> <p>The Execution Analyzer uses constraints that you define for a Class and its methods:</p> <ul style="list-style-type: none"> • A Class invariant is evaluated by the Analyzer whenever a method called on an object of the Class completes • Pre-conditions defined for a method are evaluated when the method is first called; similarly, Post-conditions are evaluated (at the same time as any Class invariant) when the method completes <p>In the Testpoint facility, constraints are composed using member and local variables in expressions separated by operands; precedence can be achieved through the use of parentheses, which also permits the construction of complex expressions - you can specify any variable that would be in scope at the time</p> <p>Elements and members of elements can also form part of the constraint; almost all equivalence operands are supported for primitive types and strings</p>	
Testpoint Constraints	<p>An <i>invariant</i> defines the rules or constraints of a Class; it is expected that no objects of the Class nor its methods can break this constraint, thus preserving the state of the object</p> <p>In the Testpoint facility, constraints are composed in much the same way as conditions are written in code; they rely on and are tied ultimately to the named attributes of the Class</p> <p>The Execution Analyzer can take a single invariant defined for a Class and test it against every invocation of any method on any instance of that Class; the results can be seen during real time in the Testpoints window</p>	The Testpoints Window ^[1676]
Trace Statements	<p>Trace statements are used to output messages during the execution of a program; the Execution Analyzer provides this facility in:</p> <ul style="list-style-type: none"> • Debugging through the use of Tracepoints and • Testing through the use of Testpoints <p>In Testpoints, a trace statement can be associated with a pre- or post- condition; whenever the condition is evaluated, the trace statement is output</p> <p>You can also group trace statements by level; when you perform a Testpoint run, you can choose the level of trace to be output</p> <p>Output can be directed to the Testpoints tab of the Output window or to a file, as configured in the Analyzer script for the parent package</p> <p>In debugging, trace statements are associated with code breakpoints</p> <p>You can include the values of variables in trace statements, by prefixing the variable name with a special token that lets the</p>	The Output Window ^[127] Managing Scripts ^[1392]

Topic	Detail	See also
	debugger know it is not a part of the text; the available tokens are: <ul style="list-style-type: none"> • \$ - when the variable is to be interpreted or printed as a string • @ - when the variable is a primitive type (int, double, char) Variables can be qualified - that is, the members of a variable can be printed You separate member names with a period character ('. '); for example: <i>The value of member c is \$a.b.c</i>	

14.5.1 Test Case



Description:

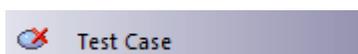
A *Test Case* is a stereotyped Use Case element. You might use it to extend the facilities of the Testing window, by applying element properties and capabilities to the tests of a feature represented by another element or - more appropriately - set of elements. That is, you can define in one go - in the Testing window for the Test Case element - the details of the tests that apply to each of several elements, instead of recording the details separately in each element.

Within the Test Case element properties you can define test requirements and constraints, and associate the test with test files. You can also link the element to Document Artifacts or (in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions) directly to linked documents, such as a Test Plan.

The Test Case element enables you to give greater visibility to tests, in the Project Browser, Diagram List, Package Browser, Model Search, Relationship Matrix, Traceability window and reports.

The Test Case element is available through the Use Case and Maintenance pages of the Toolbox.

Toolbox Icon:



14.5.2 Test Cut



Description:

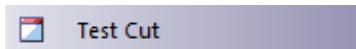
A Test Cut element is a stereotyped Object element.

A task, such as 'Print', might involve operations on different Classes. In order to create a 'Print' test, you would want to include only the 'Print' operations of these Classes and exclude any other operations.

A Test Cut enables you to capture only the operations that represent the behavior (in this case, 'Print') defined for a single Class. You might then place the Test Cut from each of several Classes into a single task as a Test Set.

When you drag a Test Cut element onto a Test Domain diagram, you create a Dependency relationship with the required Class element. As a result, when you select the *Test Cut* element on the Testpoints Window, the operations of the *Class* are listed in the window, each with a checkbox. You then select the checkbox against each Class operation to include in the Test Cut.

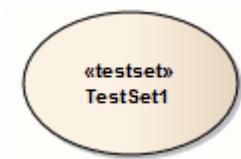
Toolbox Icon:



Learn More:

- [Test Set](#)^[1675]
- [The Testpoints Window](#)^[1676]

14.5.3 Test Set



Description:

A *Test Set* element is a stereotyped Use Case element used to aggregate one or more groups of methods (Test Cuts), which perhaps span multiple Classes, into a single task. Test Sets can also be aggregated into Test Suites.

You link the Test Cut elements to the Test Set using Dependency connectors.

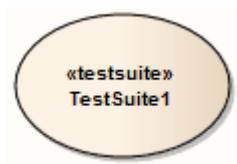
Toolbox Icon:



Learn More:

- [Test Cut](#)^[1675]
- [Test Suite](#)^[1676]

14.5.4 Test Suite



Description:

A *Test Suite* element is a stereotyped Use Case element, used to aggregate one or more groups of tasks (Test Sets).

You link the Test Set elements to the Test Suite using Dependency connectors.

Toolbox Icon:



Learn More:

- [Test Set](#)^[1675]

14.5.5 The Testpoints Window

The Testpoints window lists the test items for the testpoint object currently selected on a diagram or in the Project Browser.

You can execute, stop and filter the test runs using the window toolbar.

Access: Analyzer | Test | Testpoint Manager

Use to:

- Review tests for testpoint objects

Reference:

Column	Usage	See also
Tests	Displays the name of the selected Testpoint object and the hierarchy of objects beneath it	Test Cut ^[1675]

Column	Usage	See also
	<p>The selected object can be a:</p> <ul style="list-style-type: none"> • Class • Operation • TestCut • TestSet or • TestSuite 	<p>Test Set^[1675]</p> <p>Test Suite^[1676]</p>
Id	<p>If the test object represents an operation, an icon is displayed here when the test run commences, indicating a successful binding of that method's constraints</p>	
Invariant	<p>A pencil icon in this column indicates that a Class Invariant is defined for this Class</p> <p>Double-click on the item to add a constraint (no pencil icon) or edit the existing constraints (pencil icon) on the Class Invariant dialog</p>	<p>Class Invariant dialog^[1678]</p>
Pre	<p>A pencil icon in this column indicates that a pre-condition is defined for this method</p> <p>Double-click on the item to add a constraint (no pencil icon) or edit the existing constraints (pencil icon) on the Operation Constraints dialog</p>	<p>Operation Constraints dialog^[1679]</p>
Post	<p>A pencil icon in this column indicates that a post-condition is defined for this method</p> <p>Double-click on the item to add a constraint (no pencil icon) or edit the existing constraints (pencil icon) on the Operation Constraints dialog</p>	<p>Operation Constraints dialog^[1679]</p>
Author	<p>Displays the username of the author of the constraints</p>	
Status	<p>During a test run, indicates the last evaluated status of the test - either pass or fail</p>	
Evals	<p>During a test run, indicates the number of times the Execution Analyzer has evaluated this test</p>	
Passes	<p>During a test run, indicates the number of times the test passed</p>	
Fails	<p>During a test run, indicates the number of times the test failed</p>	
Last Run By	<p>Displays the username of the last person to run this test</p>	
Last Run Date	<p>Displays the date this test was last evaluated</p>	
Last Run Result	<p>If the test has previously been run, displays the result of the last run</p>	
Parent Collections	<p>Lists any parent collections that include the selected object as part of their design</p> <p>Double-click this collection to make it the selected object in the left pane</p>	

Learn More:

- [The Testpoints Window Toolbar](#)^[1678]

14.5.5.1 The Testpoints Window Toolbar

The Testpoint window toolbar enables you to execute, stop and filter the output of a Test run.

Access: Analyzer | Test | Testpoint Manager

Use to:

- Manage the execution of a test run

Reference:



Action	Usage	See also
Text field	Display the currently-selected test point object	
	Execute the test run	
	Stop the test run	
	Toggle between showing all items and showing only those items that have constraints	
	Toggle between showing all items and showing only operations that have been selected. This button is only enabled when a Testcut is selected. When a Testcut is selected each of the operations of its class are displayed with a checkbox. You use this checkbox to select the operations of the Testcut.	
	Display the Test Option menu, providing the following options: <ul style="list-style-type: none"> • All - Run all Tests: Include all tests and all trace statements • Testpoints Only - Run Tests only: Exclude trace statements • Tracepoints Only - Run trace statements only: Exclude tests • Trace Level menu: <ul style="list-style-type: none"> • Disabled - All trace statements are excluded • Level 1 - Level 1 trace statements are included • Level 2 - Level 1 and 2 trace statements are included • Level 3 - All trace statements are included • Trace Output menu: <ul style="list-style-type: none"> • Prefix With Function Call - Prefix all trace output lines with the executing function name 	
	Display The Testpoints Window help topic	

14.5.5.2 Class Invariant Dialog

You use the Class Invariant dialog to define or edit the invariant constraint on a Class that is subject to testpoint tests.

Access: Double-click on the Class item in the Testpoints window

Reference:

Field	Usage	See also
Name	Defaults to Invariant Leave this value or, if required, overtype it with your preferred name	
Constraints	Click on the (...) button The Testpoint Editor dialog displays, listing the conditions currently set on the Class; you can: <ul style="list-style-type: none"> list the existing conditions in a vertical list rather than a comma-separated string (Browse button) delete a condition applied to the invariant (Delete icon) create a new condition (New icon, then type the condition in) change the order in which the conditions are applied (Move Item Up, Move Item Down icons) 	
Action on Fail	Click on the drop-down arrow and select from the three options: <ul style="list-style-type: none"> Continue - ignore the failure Break execution - halt the run and display the Stack trace Disable on fail - do not execute the test again 	
OK	Click on this button to apply your changes	

Learn More:

- [The Testpoints Window](#)^[1676]

14.5.5.3 Operation Constraints Dialog

You use the Operation Constraints dialog to define or edit the pre-conditions and/or post-conditions on an operation in a Testpoint Class that is subject to testpoint tests.

Access: Double-click on the operation item in the Testpoints window

Reference:

The dialog contains two sets of the following fields; one set for pre-conditions and one set for post-conditions:

Field	Usage	See also
Constraint	Click on the (...) button The Testpoint Editor dialog displays, listing the conditions currently set on the operation; you can: <ul style="list-style-type: none"> list the existing conditions in a vertical list rather than a comma-separated string (Browse button) delete a condition applied to the invariant (Delete icon) create a new condition (New icon, then type the condition in) change the order in which the conditions are applied (Move Item Up, Move Item Down icons) 	

Field	Usage	See also
Action on Fail	Click on the drop-down arrow and select from the three options: <ul style="list-style-type: none">• Continue - ignore the failure• Break execution - halt the run and display the Stack trace• Disable on fail - do not execute the test again	
Trace	(Disabled if the Level field is set to Disabled) View the current traces and/or, if necessary, type in a new trace statement If there are several Trace statements, you can list the statements in a vertical list rather than a comma-separated string (Browse button)	
Level	Select the required level of trace statement to be output <ul style="list-style-type: none">• Disabled - no statement output• Level 1 - Only Level 1 statements output• Level 2 - Both Level 1 and Level 2 statements output• Level 3 - Levels 1, 2, and 3 statements output	

Learn More:

- [The Testpoints Window](#)^[1676]

14.6 Model Simulation

This section explains how to use the Visual Execution Analyzer to simulate the execution of behavior models. It covers the following topics:

Topic	Link
Overview of the Model Simulator	How It Works ^[1681]
How to set up simulation	Set Up Simulation ^[1681]
How to activate a simulation script	Activate a Simulation ^[1682]
How to run the model simulation	Run Model Simulation ^[1683]
How to set up and use breakpoints	Using Simulation Breakpoints ^[1684]

14.6.1 How It Works

The Model Simulator is a tool in Enterprise Architect's Visual Execution Analyzer, enabling you to simulate the execution of conceptual model designs containing behavior. It provides a quick and simple way to verify your design's behavior for logical correctness.

Topics:

Topic	Detail	See also
Platforms Supported	<p>The Model Simulator supports the following simulation platform:</p> <ul style="list-style-type: none"> UML Basic <p>The Model Simulator currently supports the execution of Activity, Interaction and State Machine models</p>	Set Up Simulation ^[1681]

Learn More:

- [Set Up Model Simulation](#) ^[1681]
- [Run Model Simulation](#) ^[1683]

14.6.2 Set Up Simulation

Setting up a model for simulation is a simple process. An Execution Analyzer script is configured for the package containing the required simulation model.

To configure a simulation execution script, you must select the package on the Project Browser, Package Browser, Diagram List or Model Search.

Access: [Project Browser Package Context Menu | Execution Analyzer | Analyzer Scripts...](#)

How To:

To set up a model for simulation, follow the steps below:

Step	Action	See Also
1	In the Execution Analyzer window, click on the New toolbar icon (second from the left)	
2	Select the required package in the Browse Project dialog and click on the OK button	
3	In the Execution Analyzer dialog, select the Simulation page	
4	In the Entry Point field, click on the (...) button and select the required entry point for the simulation, and the required Activity, Interaction or State Machine to simulate	
5	In the Platform field, click on the drop-down arrow and select UML Basic	
6	If required, select the Record Execution checkbox to record the simulation session	Control the Recording Session <small>1648</small>
7	Click on the OK button to save changes	

Notes:

- If you do not specify an entry point, the simulator attempts to simulate the entire package
- All simulation elements and relationships must reside within the package configured for simulation

Learn More:

- [How Simulation Works](#) 1681
- [Activate Simulation Script](#) 1682
- [Run Model Simulation](#) 1683
- [Using Simulation Breakpoints](#) 1684

14.6.3 Activate Simulation Script

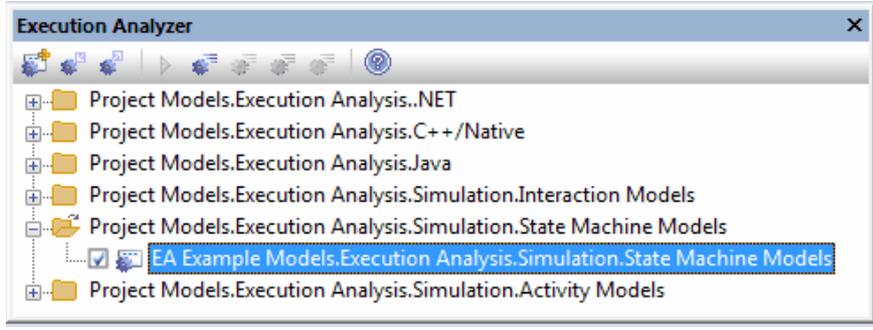
An Execution Script is configured for a model package defining the simulation parameters. To run the simulation you must first activate the execution script on the Execution Analyzer window.

Access: [Project Browser Package Context Menu | Execution Analyzer | Analyzer Scripts...](#)

How to:

To activate a simulation script for execution, follow the steps below:

Step	Action	See Also
1	In the Execution Analyzer window, select the required execution script	

Step	Action	See Also
		
2	Click on the checkbox to the left of the script to activate it	
3	Execute the simulation using the Analyzer Simulation menu option	Run Model Simulation ^[1683]

Notes:

- The simulation tool only becomes active when a valid simulation execution script is activated

Learn More:

- [Set Up Simulation](#) ^[1681]

14.6.4 Run Model Simulation

You run a model simulation through the Simulation window, by executing the activated simulation script configured for the model package.

Access: **Analyzer | Simulation > Output**

Topics:

Topic	Detail	See also
Abstract	<p>The simulation executes the model step-by-step, enabling you to validate the logic of your behavior model; the simulation automatically highlights the execution step in the model's diagram</p> <p>If a branch is encountered in the execution, the simulator prompts you to choose the appropriate path to take in your execution</p> <p>The Simulation window toolbar icons are explained in the table below</p>	Activate Simulation Script ^[1682]

Icon	Action	See Also
	Start the simulator using the activated simulation script	Activate Simulation Script ^[1682]
	Pause the simulation	

Icon	Action	See Also
	When the simulation is paused, step over, step in and step out to control the simulator's execution at the required step in the model simulation	
Slider	Vary the runtime speed of the simulation	
	Stop the simulation	

Notes:

- The Simulation tool only becomes active when a valid simulation Execution Script is activated

Learn More:

- [Set Up Simulation](#)^[1681]
- [Using Simulation Breakpoints](#)^[1684]

14.6.5 Using Simulation Breakpoints

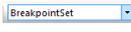
The Simulation Breakpoints window enables you to apply control to the Model Simulation process.

Access: **Analyzer | Simulation > Breakpoints**

Topics:

Topic	Detail	See also
Abstract	<p>The simulation executes the model step-by-step, enabling you to validate the logic of your behavior model; the simulation halts when it reaches an element defined as a breakpoint</p> <p>The UML elements that can be defined as breakpoints include: <i>Actions</i>, <i>Activities</i>, and <i>States</i></p> <p>The UML relationships that can be defined as breakpoints include: <i>Interaction Messages</i></p> <p>The breakpoints are stored as Breakpoint Sets for a given Enterprise Architect project</p>	Run Model Simulation ^[1683]

Icon	Action	See Also
	Enables all breakpoints defined in the current Breakpoint Set for the simulation session	
	Deletes all breakpoints defined in the current Breakpoint Set for the simulation session	
	Disables all breakpoints defined in the current Breakpoint Set for the simulation session	
	Adds a breakpoint for the selected element or Sequence message to the current Breakpoint Set	

Icon	Action	See Also
	Changes the selected Breakpoint Set for use in the simulation session	
	Performs Breakpoint Set commands: <ul style="list-style-type: none">• New Set: Create a new Breakpoint Set• Save As Set: Saves the current Breakpoint Set under a new name• Delete Selected Set: Deletes the current Breakpoint Set• Delete All Sets: Deletes all Breakpoint Sets saved for the project	

Learn More:

- [Run Model Simulation](#)^[1683]

Part



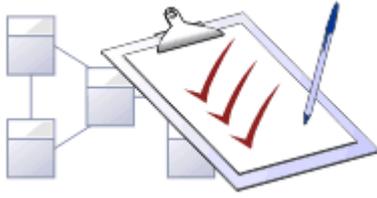
15 Testing



This section discusses quality control in your project with Enterprise Architect, describing:

Topic	Link
Model Validation Checks - Used to check UML elements, diagrams or packages against known UML rules (identified in configuring validation) and constraints defined within the model, using the Object Constraint Language (OCL)	Model Validation Checks ^[1688]
Testing - Enterprise Architect enables you to create and manage test scripts for model elements, covering unit, integration, scenario, system and acceptance tests	Testing ^[1694]
Testpoint Management - provides the facility to pass or fail application tasks, viewing test results in real time as the program executes and results are saved	TestPoint Management ^[1672]
Integration with the unit testing tools JUnit and NUnit	Unit Testing ^[1669]

15.1 Model Validation



You use Model Validation to check UML models against known **UML rules** (which you identify in **configuring validation**) as well as any constraints defined within the model, using the Object Constraint Language (OCL). You can run Model Validation against a single UML element, a diagram or an entire package.

Access: **Project | Model Validation | Validate Selected (Ctrl + Alt + V)**

Topics:

Topic	Detail	See also
Validating	Validating a UML: <ul style="list-style-type: none"> • Element validates the element and its children, features (attributes and operations) and relationships (connectors) • Diagram validates the diagram itself (for correctness) as well as any elements and connectors within the diagram • Package validates the package and all sub-packages, elements, connectors and diagrams within it 	
Example - Model Violation	<p>The following UML diagram contains several basic violations:</p> <p>If you run Model Validation on this diagram, Enterprise Architect displays the following violations in the Output window:</p> <ul style="list-style-type: none"> • It contains a UML ExpansionRegion (<i>ExpansionRegion1</i>) that 	

Topic	Detail	See also
	<p>is missing its child input ExpansionNode</p> <ul style="list-style-type: none"> • It contains an invalid self-generalization on <i>Class2</i> (UML elements cannot be self-generalized) • It contains an OCL violation for the anonymous Association (between <i>Class2</i> and <i>Object1</i>) • It contains a UML ExceptionHandler (<i>ExceptionHandler1</i>) that is missing its child input ObjectNode 	

How To:

To use Model Validation, follow the steps below:

Step	Action	See Also
1	Select the package, diagram or element either from the Project Browser or within an open diagram	
2	<p>Select the Validate Selected menu option</p> <p>Enterprise Architect performs the validation, and displays the results in the Output window (if the Output window does not automatically display, select the View System Output menu option)</p> <p>While performing the validation, Enterprise Architect also displays a progress window containing the Cancel Validation button, which enables you to cancel the validation process at any time</p> <p>Alternatively, select the Project Model Validation Cancel Validation menu option</p>	Output Window ^[127]

Notes:

- If you double-click on an error in the Output window, you select the diagram element that the error message refers to

Learn More:

- [UML rules](#)^[1690]
- [Configuring Model Validation](#)^[1689]

15.1.1 Configure Model Validation

Use the Model Validation Configuration dialog to enable and disable the rules that are run with the model validator. You can define additional rules in this dialog from any additional Add-Ins that might be installed besides Enterprise Architect.

Access: **Project | Model Validation | Configure**

Topics:

Topic	Detail	See also
Usage	<p>Click on the checkbox against each Validation Rule to apply in performing a model validation</p> <p>When you perform a validation, each violation listed on the Output window has a violation ID of the format <i>MVRxxnnnn</i>, where:</p> <ul style="list-style-type: none"> <i>MVR</i> stands for Model Validation Rule <i>xx</i> is a hexadecimal number corresponding to the position of the validation rule in the Model Validation Configuration dialog, thus indicating which rule is applied and violated <i>nnnn</i> is the number of the violation message <p>Therefore:</p> <ul style="list-style-type: none"> Messages with the ID <i>MVR01nnnn</i> indicate that the Element: Well-Formedness checkbox is selected and a violation of that rule has been detected Messages with the ID <i>MVR0Annnn</i> indicate that the Feature: OCL Conformance checkbox (10th in order on the dialog, or <i>Ath</i> in hexadecimal) is selected and a violation of that rule has been detected 	Model Validation ¹⁶⁸⁸

Notes:

- To disable UML syntax ("*The requested connection is not UML compliant*"), select the **Tools | Options | Diagram** option, and in the General panel deselect the **Strict UML Syntax** checkbox

Learn More:

- [Rules Reference](#) ¹⁶⁹⁰

15.1.2 Rules Reference

Topics:

Topic	Detail	See also
Validation Rules	<p>Model Validation works against a set of validation rules, arranged in the following groups:</p> <ul style="list-style-type: none"> (Element, Relationship, Feature, Diagram): Well-Formedness Checks whether or not an element, relationship, feature or diagram is well-formed; this group of rules includes checks such as whether the item is a valid UML item and whether a diagram contains valid elements within it Element: Composition Checks whether or not a UML element contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children (Element, Relationship, Feature): Property Validity Checks whether or not the item has the correct UML properties defined, and whether the properties contain incorrect or conflicting values (Element, Relationship, Feature): Custom Properties Validates an item against any defined constraints in OCL 	Well-Formedness ¹⁶⁹¹ Element: Composition ¹⁶⁹¹ Property Validity ¹⁶⁹² OCL Conformance ¹⁶⁹²

15.1.2.1 Well-Formedness

This group of rules checks whether or not an element, relationship, feature or diagram is well-formed. The rules includes checks such as whether the item is a valid UML item and whether a diagram contains valid elements within it.

Reported violations:

Violation ID	Description	Information
MVR010001	«Element» is not a valid UML Element	The element is not a recognized UML 2.3 element
MVR050001	«Relationship» is not a valid UML Relationship	The relationship is not a recognized UML 2.3 relationship
MVR050002	«Relationship» is not legal for «Start Element» --> «End Element»	The relationship between the given start and end elements is not valid for those elements
MVR050003	«Parent Element»:isLeaf=true and cannot be generalized by «Child Element»	The Generalization relationship cannot exist between parent and child elements because the parent element is defined as a leaf element
MVR050004	«Child Element»:isRoot=true and cannot generalize «Parent Element»	The Generalization relationship cannot exist between parent and child elements because the child element is defined as a root element
MVR050005	«Element» cannot generalize self	The element cannot be self-generalized
MVR0B0001	Statechart violation: «extended information»	The State diagram contains a UML violation; see the extended information for more information about the detected violation

15.1.2.2 Element Composition

This group of rules checks whether or not a UML element contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children.

Reported violations:

Error ID	Description	Information
MVR020001	«Element» is missing required child element «Child Element»	The element is missing a child element of type <i>Child Element</i>
MVR020002	Invalid UML package child	The element cannot be a direct package child and must be a child of another element (for example: Ports must be children of other elements, and not direct UML package members)
MVR020003	Invalid child «Child Element name» («Child Element Type»)	The child element is invalid on the tested parent element

15.1.2.3 Property Validity

This group checks whether or not an element, relationship or feature has the correct UML properties defined for it and whether they contain incorrect or conflicting values.

Reported violations:

Error ID	Description	Information
MVR030001	«Element»:«Property» property is undefined	The element property contains no value
MVR030002	«Element»:«Property» property has invalid value: "«Value»"	The element property contains an invalid value
MVR030003	«Element»:isLeaf=true and cannot be abstract	The element's <i>isLeaf</i> and <i>isAbstract</i> properties are both set to true , which is invalid
MVR060001	«Relationship»:«Property» property is undefined	The relationship property contains no value
MVR060002	«Relationship»:«Property» property has invalid value: "«Value»"	The relationship property contains an invalid value
MVR090001	Attribute/AssociationEnd mismatch, «Attribute»: «Mismatch description», ...	The given attribute has an <i>associationEnd</i> of the same name but they differ in the listed details

Learn More:

- [Properties Dialog](#)^[660] (Element)
- [Connector Properties](#)^[756]

15.1.2.4 OCL Conformance

This group validates an element, relationship or attribute against any defined constraints in the Object Constraint Language (OCL). OCL is used to describe expressions on UML models, and to express constraints free of side-effects. You can add OCL constraints to any element, relationship or attribute in Enterprise Architect.

Reported violations:

Error ID	Description	Information
MVR040001	OCL violation: «violated OCL»	The element violates the OCL constraint specified
MVR070001	OCL violation: «violated OCL»	The relationship violates the OCL constraint specified
MVR0A0001	OCL violation: «violated OCL»	The attribute violates the OCL constraint specified

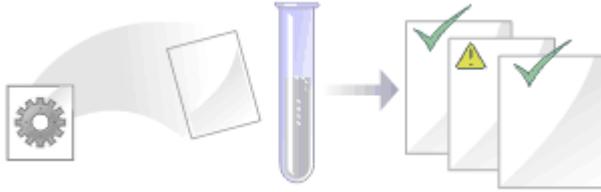
Topics:

Topic	Detail	See also
Define OCL Constraints for an Element	<p>You can add OCL constraints to an element using the Properties dialog (Element Properties); select the Constraints page, click on the Type drop-down arrow and select OCL</p> <p>To perform an OCL Validation, display the Model Validation Configuration dialog and select the Element: (OCL) Conformance checkbox; any OCL violations are recorded in the Model Validation tab of the Output window</p>	Model Validation Configuration Model Validation
Define OCL Constraints for a Relationship	<p>You can add OCL constraints to a relationship using the Properties dialog (right-click and select the <type> Properties context menu option); select the Constraints page, click on the Type drop-down arrow and select OCL</p> <p>To perform an OCL Validation, display the Model Validation Configuration dialog and select the Relationship: (OCL) Conformance checkbox; any OCL violations are recorded in the Model Validation tab of the Output window</p>	
Define OCL Constraints for an Attribute	<p>You can add OCL constraints to a feature using the Properties dialog (Element Attributes); select the Constraints tab, click on the Type drop-down arrow and select OCL</p> <p>To perform an OCL Validation, display the Model Validation Configuration dialog and select the Feature: (OCL) Conformance checkbox; any OCL violations are recorded in the Model Validation tab of the Output window</p>	

Notes:

- To have a valid OCL constraint, the syntax must be correctly formed; if the expression is not correct, Enterprise Architect displays a message stating that the OCL constraint is not valid

15.2 Testing



Topics:

Topic	Detail	See also
Introduction to Testing	<p>Enterprise Architect enables you to create and manage test scripts for model elements, covering unit, integration, scenario, system and acceptance tests; this can include test cases generated from xUnit testing and Testpoint Management</p> <p>You can import tests from other elements, generate them from scenarios, and generate test documentation and reports; you can also indicate the presence of tests on an element by displaying test information on the element in a diagram</p> <p>Enterprise Architect enables you to attach arbitrarily complex tests to any model element; keeping the model elements and the testing documentation in one integrated model significantly improves the communication between the test-team and the software developers and architects</p> <p>The detailed search facilities make it easy to find failing test cases, test cases not run and test cases that have been passed; using the testing and search capabilities, it is easy to navigate through the model and quickly locate problem spots, design flaws and other critical issues</p> <p>Enterprise Architect is not only a UML Modeling environment, it is also a complete Test Management environment</p>	xUnit Testing ^[1669] Testpoint Management ^[1672]
Basic Tasks	<p>Simple tasks that you might perform include:</p> <ul style="list-style-type: none"> • Open the Testing Workspace • Use the Test Details dialog 	The Testing Workspace ^[1695] The Test Details Dialog ^[1696]
Categories	<p>Typically, you create:</p> <ul style="list-style-type: none"> • Unit tests for things that are being built, such as Classes and components • Integration tests to test how components work together • System tests to ensure the system meets business requirements • Acceptance tests to test user satisfaction, and • Scenario tests to test the end-to-end suitability and functionality of the application 	Unit Tests ^[1696] Integration Testing ^[1698] System Testing ^[1699] Acceptance testing ^[1700] Scenario testing ^[1701]
Using Tests	<p>Other tasks that you might perform when working with tests include:</p> <ul style="list-style-type: none"> • Import a scenario as a test • Move or copy tests between test types • Import a test from other elements 	Import a Scenario as a Test ^[1703] Move or Copy Tests Between Test Types ^[1703]

Topic	Detail	See also
	<ul style="list-style-type: none"> • Import a responsibility or constraint as a test • Create a maintenance item from a test • Generate a Test Details report • Show test script compartments • Create test documentation 	Import a Test from Other Elements ^[1705] Import a Responsibility or Constraint as a Test ^[1706] Create a Maintenance Item from a Test ^[1707] Generate Test Details Report ^[1786] Show Test Script Compartments ^[1707] Create Test Documentation ^[1708]

Notes:

- Most of the tasks identified above relate to a tests for a single element
- You can make a set of tests available to a number of elements by performing the above tasks on a **Test Case** element and then associating that Test Case with each of the other elements; the Test Case element also helps to make tests more visible in diagrams, the Project Browser, windows and searches

Learn More:

- [Test Case](#)^[1673]

15.2.1 The Testing Workspace

The Testing window, or Workspace, provides a quick and convenient method of working with element tests. When you select an element in a diagram or in the Project Browser, if the Testing window is visible the lists of tests for that element are loaded ready for modification or addition.

Access: [View](#) | [More Element Tools](#) | [Testing](#) (Alt + 3)

Topics:

Topic	Detail	See also
Using the Testing Workspace	<p>The window has two formats - Item mode and List mode respectively</p> <p>To toggle between the modes, click on the Show/Hide Properties button in the window toolbar</p> <p>To add new items, click on the New icon in the window toolbar; in:</p> <ul style="list-style-type: none"> • Item mode, this clears the fields for new data • List mode, this displays the Test details dialog <p>By clicking on the Auto button in Item mode or on the Test details dialog, you can apply an automatic naming/numbering nomenclature that you have previously defined</p>	Test Dialog ^[1696] Use Auto Naming and Auto Counters ^[628] Unit testing ^[1696] Integration testing ^[1698] System testing ^[1699] Acceptance testing ^[1700]

Topic	Detail	See also
	<p>There are five tabs along the base of the window; one for each of the following types of testing:</p> <ul style="list-style-type: none"> • Unit testing • Integration testing • System testing • Acceptance testing • Scenario testing <p>The asterisk on a tab indicates that the tab contains saved information; if there is no information for a type of test, or the information has not yet been saved, its tab has no asterisk</p> <p>The tabs toggle between Item mode and List mode independently</p>	<p>Scenario testing^[170]</p>

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records
- The Testing window can be docked to the application workspace

Learn More:

- [Permission List](#)^[205]

15.2.2 The Test Details Dialog

The Test Details dialog opens to edit tests when the Testing window is in List mode.

Double-click on an existing test case or click on the **New** icon in the window toolbar. The Test Details dialog displays.

Access: **View | More Element Tools | Testing (Alt + 3)**

Notes:

- Add multiple test cases in one batch using the **New** and **Apply** buttons
- You can format the text in the Description, Input, Acceptance Criteria and Results tabs using the Notes toolbar at the top of the field; this text is also reflected in the Notes window, but cannot be edited there
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

Learn More:

- [Notes Toolbar](#)^[77]

15.2.3 Unit Testing

Use Unit Testing to test Classes, Components and other elements as programmers build them.

Access: **View | More Element Tools | Testing > Unit**

The Unit testing tab displays in the Testing window by default.

If any **Unit tests** exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference:

Field	Usage	See also
Test	Specify the name of the test	
Status	Click on the drop-down arrow and select the current status of the test	
Type	Click on the drop-down arrow and select the type of test	
Run By	Click on the drop-down arrow and select the name of the person who ran the test	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run	
Last Run	Click on the drop-down arrow and select the date on which the test was last run	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field This text is also reflected in the Notes window, but cannot be edited there	Notes Toolbar ^[770]
Input	Type in the input data	
Acceptance Criteria	Type the acceptance or test success conditions	
Results	Type the results of the last test	

Topics:

Topic	Detail	See also
Editing Unit Tests	To edit existing Unit Test items for this element, either <ul style="list-style-type: none"> • Click on the item in the left-hand panel in Item mode • Double-click on the item in List mode to display the Test Details dialog, or • Click on the required item in the <i>Testing</i> folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); Unit Test item icons have a U in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser</p>	Test Details Dialog ^[696] Element Browser ^[687]

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

15.2.4 Integration Testing

Use Integration Testing to test how the constructed components work together.

Access: [View | More Element Tools | Testing > Integration](#)

Open a diagram or the Project Browser and select the required element, then click on the Integration tab.

If any Integration tests exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference:

Field	Usage	See also
Test	Specify the name of the test	
Status	Click on the drop-down arrow and select the current status of the test	
Type	Click on the drop-down arrow and select the type of test	
Run By	Click on the drop-down arrow and select the name of the person who ran the test	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run	
Last Run	Click on the drop-down arrow and select the date on which the test was last run	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field This text is also reflected in the Notes window, but cannot be edited there	Notes Toolbar ^[770]
Input	Type in the input data	
Acceptance Criteria	Type the acceptance or test success conditions	
Results	Type the results of the last test	

Topics:

Topic	Detail	See also
Editing Integration Tests	To edit existing Integration Test items for this element, either <ul style="list-style-type: none"> • Click on the item in the left-hand panel in Item mode • Double-click on the item in List mode to display the Test Details dialog, or • Click on the required item in the Testing folder in the 	Test Details Dialog ^[1698] Element Browser ^[687]

	<p>Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); Integration Test item icons have a I in the bottom right corner</p> <p>To change the element for which to create test items, click on the required element in the Project Browser</p>	
--	--	--

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

15.2.5 System Testing

Use System Testing to test that the system performs the right business functions correctly.

Access: [View](#) | [More Element Tools](#) | [Testing](#) > [System](#)

Open a diagram or the Project Browser and select the required element. Click on the System tab.

If any System Tests exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference:

Field	Usage	See also
Test	Specify the name of the test	
Status	Click on the drop-down arrow and select the current status of the test	
Type	Click on the drop-down arrow and select the type of test	
Run By	Click on the drop-down arrow and select the name of the person who ran the test	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run	
Last Run	Click on the drop-down arrow and select the date on which the test was last run	
Description	<p>Type a description of the test; you can format the text using the Notes toolbar at the top of the field</p> <p>This text is also reflected in the Notes window, but cannot be edited there</p>	Notes Toolbar 17701
Input	Type in the input data	
Acceptance Criteria	Type the acceptance or test success conditions	
Results	Type the results of the last test	

Topics:

Topic	Detail	See also
Editing System Tests	<p>To edit existing System Test items for this element, either</p> <ul style="list-style-type: none"> • Click on the item in the left-hand panel in Item mode • Double-click on the item in List mode to display the Test Details dialog, or • Click on the required item in the <i>Testing</i> folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); System Test item icons have a Sy in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser</p>	<p>Test Details Dialog^[1696]</p> <p>Element Browser^[687]</p>

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

15.2.6 Acceptance Testing

Use Acceptance Testing to ensure that users are satisfied with the system.

Access: [View | More Element Tools | Testing > Acceptance](#)

Open a diagram or the Project Browser and select the required element. Click on the Acceptance tab.

If any Acceptance Tests exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference:

Field	Usage	See also
Test	Specify the name of the test	
Status	Click on the drop-down arrow and select the current status of the test	
Type	Click on the drop-down arrow and select the type of test	
Run By	Click on the drop-down arrow and select the name of the person who ran the test	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run	
Last Run	Click on the drop-down arrow and select the date on which the test was last run	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field	Notes Toolbar ^[770]

Field	Usage	See also
	This text is also reflected in the Notes window, but cannot be edited there	
Input	Type in the input data	
Acceptance Criteria	Type the acceptance or test success conditions	
Results	Type the results of the last test	

Topics:

Topic	Detail	See also
Editing Acceptance Tests	<p>To edit existing Acceptance Test items for this element, either</p> <ul style="list-style-type: none"> Click on the item in the left-hand panel in Item mode Double-click on the item in List mode to display the Test Details dialog, or Click on the required item in the Testing folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); Acceptance Test item icons have a A in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser</p>	<p>Test Details Dialog ^[1696]</p> <p>Element Browser ^[687]</p>

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

15.2.7 Scenario Testing

Use Scenario Testing to test the application with real-world situations and scenarios. An end-to-end test of all functions.

Access: [View](#) | [More Element Tools](#) | [Testing > Scenario](#)

Open a diagram or the Project Browser and select the required element. Click on the Scenario tab.

If any Scenario Tests exist, the first Test item for the element is shown in the Testing window in Item mode, and all items are listed either in the panel to the left of the window, or in the window in List mode.

Reference:

Field	Usage	See also
Test	Specify the name of the test	

Field	Usage	See also
Status	Click on the drop-down arrow and select the current status of the test	
Type	Click on the drop-down arrow and select the type of test	
Run By	Click on the drop-down arrow and select the name of the person who ran the test	
Checked By	Click on the drop-down arrow and select the name of the person who checked the test run	
Last Run	Click on the drop-down arrow and select the date on which the test was last run	
Description	Type a description of the test; you can format the text using the Notes toolbar at the top of the field This text is also reflected in the Notes window, but cannot be edited there	Notes Toolbar ^[770]
Input	Type in the input data	
Acceptance Criteria	Type the acceptance or test success conditions	
Results	Type the results of the last test	

Topics:

Topic	Detail	See also
Editing Scenario Tests	To edit existing Scenario Test items for this element, either <ul style="list-style-type: none"> Click on the item in the left-hand panel in Item mode Double-click on the item in List mode to display the Test Details dialog, or Click on the required item in the Testing folder in the Element Browser window (if this window is not displayed, click on the  icon in the Testing window toolbar); Scenario Test item icons have a Sc in the bottom right corner <p>To change the element for which to create test items, click on the required element in the Project Browser</p>	Test Details Dialog ^[1696] Element Browser ^[687]

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

15.2.8 Move or Copy Tests Between Categories

When you define a test on the Unit, Integration, System, Acceptance or Scenario tab of the Testing window, you might decide that the test either is better suited to another category of tests, or forms a good template for tests in other categories. Enterprise Architect enables you to move or copy tests between categories.

How To:

To move or copy a test, follow the steps below

Step	Action	See Also
1	Open either: <ul style="list-style-type: none"> The Testing window, and the tab that contains the test you want to move or copy The <i>Testing</i> folder of the Element Browser window 	Element Browser ^[68]
2	Right-click on the required test item in the list The item context menu displays	
3	Click on the appropriate option - Move to or Copy to A list of test categories displays	
4	Click on the test category to which to move or copy the test A confirmatory prompt displays	
5	Click on the Yes button to confirm the move or copy	
6	Click on the target tab of the Testing window to ensure that the test has been added, and make any required changes	

Notes:

- If you move or copy a test into the Scenario category, some unassociated data could be lost

15.2.9 Import Scenario as Test

You can import a scenario from a Use Case or other element, or from all elements in a package, into the Testing window Scenario tab. This avoids having to duplicate the scenario information manually.

The scenario description is copied to the Scenario Test Description tab in the Testing window. If a scenario contains a Structured Specification, its **Action** steps are also copied to the Description tab under the heading **Structured Specification**.

How To:

To import a scenario from a single element, follow the steps below:

Step	Action	See Also
1	Open a diagram or the Project Browser and select the element into which to import the scenario	

Step	Action	See Also
2	Open either: <ul style="list-style-type: none"> The Testing window and the Scenario tab The <i>Testing</i> folder of the Element Browser window 	Element Browser ^[687]
3	Right-click on the list of tests to display the context menu, and select the Import element scenario(s) menu option The Import Scenario dialog displays	
4	You can import scenarios from any element in the model by clicking on the Select element drop-down arrow and selecting the required element Select the scenarios to import from the Select items to import list	
5	Click on the OK button to import the selected scenario(s)	

To import scenarios from all elements in a package, follow the steps below:

Step	Action	See Also
1	Open a diagram or the Project Browser and select the parent package element or an element within the package	
2	Open either: <ul style="list-style-type: none"> The Testing window and the Scenario tab The <i>Testing</i> folder of the Element Browser window 	Element Browser ^[687]
3	Right-click on the list of tests and select the Import Package Scenario(s) context menu option The Import Scenario dialog displays This version of the Import Scenario dialog lists all scenarios against all elements in the package; it does not enable you to select a specific element, but does enable you to filter the list of scenarios to those from specific types of element	
4	In the Limit selection to these Object Types only field, type a comma-separated list of the object types for which to show scenarios Click on the Refresh button	
5	Click on the OK button to import the scenarios from each element as test scenarios for that element	

Reference:

The Import Scenario dialog has the following additional options:

Field	Usage	See also
Show related elements only	Filter selection to apply only to related elements	
Limit selection to	Indicates specific element types, separated by commas, to filter for	

Field	Usage	See also
these Object Types only	only those element types	
Refresh	Refresh the list of available scenarios	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

15.2.10 Import Test From Other Elements

You can import any test from a Use Case or other element into the Testing window. This avoids having to duplicate the test information manually.

How To:

To import a test, follow the steps below:

Step	Action	See Also
1	Open a diagram or the Project Browser and select the element into which to import the tests	
2	Open either: <ul style="list-style-type: none"> The Testing window, or The <i>Testing</i> folder of the Element Browser window 	Element Browser
3	Right-click on the list of tests and select the Import Tests from Other Element context menu option The Import Element Tests dialog displays	
4	You can import tests from any element in the model by clicking on the Select element drop-down arrow and selecting the required element Select the test to import from the Select items to import list	
5	Click on the OK button to import the selected test(s)	

Reference:

The Import Element Tests dialog has the following additional options:

Option	Action	See also
Show related elements only	Filter selection to apply only to related elements	
Limit Selection to these Object Types only	Indicates specific element types, separated by commas, to filter for only those element types	

Option	Action	See also
Refresh	Refresh available options	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

15.2.11 Import Responsibility or Constraint as Test

In the same way as you can import a scenario from an element as a test, you can also import an internal requirement (responsibility) or internal constraint as a test.

How To:

To import a requirement or constraint as a test, follow the steps below:

Step	Action	See Also
1	Open a diagram or the Project Browser and select the element into which to import the responsibility or constraint	
2	Open either: <ul style="list-style-type: none"> The Testing window and the tab into which to import the test The <i>Testing</i> folder of the Element Browser window 	Element Browser [687]
3	Right-click on the list of tests to display the context menu (if in the Element Browser, click on a test of the appropriate type)	
4	Click on the appropriate option, either: <ul style="list-style-type: none"> Import element constraint(s) Import element requirement(s) <p>The Import Constraint or Import Requirements dialog displays (the two dialogs are identical)</p> <p>The dialog lists all of the internal requirements or constraints in the selected element</p>	
5	Select one or more of the items to import as tests, and click on the OK button	
	Each item is added to the list of tests in the Testing window, as a standard, 'Not Run' test	
6	Edit the items to complete their definition as tests	

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

15.2.12 Create Maintenance Item From Test

If an element fails a test, one likely consequence is that a Defect (Issue) item has to be raised in model maintenance to correct the problem. You can generate this Defect item directly from the test that failed.

How To:

To create a Maintenance item from a test, follow the steps below:

Step	Action	See Also
1	In the Testing window, or the <i>Testing</i> folder of the Element Browser window, right-click on the test item from which to generate the Maintenance item	
2	On the context menu, select the Create a Maintenance Defect from this test menu option A message box displays	
3	Click on the OK button to clear the message	
4	Open the Maintenance window and select the Defects tab The window displays a defect item having the same name as the test, and the test Description , Input , Acceptance Criteria and Results texts are displayed in the defect Description tab	
5	Update the defect item as required	Maintenance Item Properties 

Notes:

- You can create Maintenance Defect items from several Test items at once; press and hold (**Shift**) as you select the Test items, and then right-click and proceed as above - each selected Test item then generates a Defect item

15.2.13 Show Test Script Compartments

Any element that is capable of displaying a compartment can be used to show test scripts in a diagram. To make use of the feature the element must have an attached test.

How To:

To show the Test Script compartment on an element in a diagram, follow the steps below:

Step	Action	See Also
1	Open a diagram containing the element with the attached test items	
2	Double-click on the diagram background to display the Diagram Properties dialog Click on the Elements tab	
3	In the Show Compartments panel, select the Testing checkbox	
4	Click on the OK button to save the setting The tests now appear as an item in the test scripts compartment of the diagram	

Step	Action	See Also
	element	

Example:

CCBookClientPanel		JPanel
~	bookpanel: CCBookPanel = new CCBookPanel()	
~	lbl_Price: JLabel = new JLabel("Price")	
+	CCBookClientPanel()	
+	GetBook() : CBook	
-	getUI() : void	
-	jbInit() : void	
+	SetBook(CBook) : void	
-	setUI() : void	
test scripts		
Acceptance: : (Pass) User Satisfaction		

15.2.14 Test Documentation

Enterprise Architect enables you to output the test scripts and results you have entered against elements in the model, in Rich Text format. For more information on entering test scripts and details see the previous sections of the Testing topic.

Access: **Project Browser Package context menu | Documentation | Testing Report**

Notes:

- You can also access the Generate Test Documentation dialog by selecting the **Project | Documentation | Testing Report** menu option
- The Generate Test Documentation dialog enables you to set up your report; you can configure which tests to include or exclude in the report, whether to include child packages and what file to output to

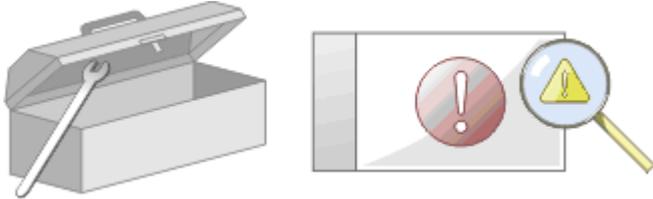
Learn More:

- [Testing](#)^[1694]

Part



16 Maintenance



This section discusses the Enterprise Architect facilities for:

- Creating and monitoring maintenance items on project elements
- Creating and monitoring change and issue items on the project

Learn More:

- [Maintenance Items](#) [1713]
- [Changes and Defects](#) [1719]

16.1 Maintenance Diagram

A Maintenance diagram is a custom diagram used to describe change requests and issue items within a system model.

Change, Task and Issue elements can be linked back to other model elements in the system to illustrate how they must be modified, fixed or updated.

Maintenance models provide extensions to the UML model and enable change management of change items, and of the model elements that require the changes to be made to them.

Example Diagram: [Example Maintenance Diagram](#) 

Tools:

Select Maintenance diagram elements and connectors from the Maintenance pages of the Toolbox.

Maintenance Diagram Elements	Maintenance Diagram Connectors
 Package	 Aggregate
 Issue	
 Change	
 Test Case	
 Entity	

16.1.1 Example Maintenance Diagram

An example Maintenance diagram is shown below.

Changes

EA supports custom Elements of type 'Change'. These can be linked to other elements in the repository or used as a separate lists of any changes proposed for the model.

Below are a set of Change elements for the shopping basket with test definitions listed against them. They are ready for checking once they are confirmed as corrected.

The color markings reflect the Status of the element.

View Basket - add: alter quantity against the entries.	□
test scripts	
Unit: : (Not Run) Alter Quantity	

View Basket - Add button to update Change	□
test scripts	
Unit: : (Deferred) Update Cart Button operative	

Create Account: password confirmation fails	□
test scripts	
Unit: : (Not Run) Password Confirmation	

16.2 Maintenance



Topics:

Topic	Detail	See also
Maintenance Items	<p><i>Maintenance</i> items are defects, changes, issues and tasks. They all apply to individual model elements and can be used to record and capture problems, changes, issues and tasks as they arise, and document the solution and associated details. They are defined as follows:</p> <ul style="list-style-type: none"> • A defect can be considered as a failure to meet a requirement for the current model element • A change can be considered as a change in requirement for the current model element • An issue records a risk factor that might affect the project being recorded for the current model element • A task is a means of recording work in progress and work outstanding for the current model element <p>Note that each of these maintenance items applies at the model element level. For <i>changes</i>, <i>defects</i> and <i>issues</i> that apply to the whole system, see the Changes and Defects topic; for tasks that apply to the whole system, see the Project Tasks topic.</p>	<p>Changes and Defects ^[1713]</p> <p>Project Tasks ^[356]</p>
Creating and Editing Maintenance Items	<p>The following topics explain how to create and edit Maintenance items:</p> <ul style="list-style-type: none"> • The Maintenance Workspace - describes the Maintenance window • Maintenance Item Properties - describes how to complete the Maintenance window tabs for the various maintenance items • Move or Copy Maintenance Items - describes how to move items between maintenance categories or generate items from an item in a different category • Create Elements From Maintenance Items - describes how to generate elements from maintenance items • Show Maintenance Script in Diagram - describes how to display maintenance items in elements on diagrams • Insert Maintenance Feature - describes how to add a maintenance item directly to an element via in-place editing 	<p>The Maintenance Workspace ^[1713]</p> <p>Maintenance Item Properties ^[1715]</p> <p>Move or Copy Maintenance Items ^[1716]</p> <p>Create Elements From Maintenance Items ^[1717]</p> <p>Insert Maintenance Feature ^[727]</p>

16.2.1 The Maintenance Workspace

Enterprise Architect makes it easy to record and capture problems and issues as they arise, and document the solution and associated details. The **Maintenance** window provides a quick method of viewing and modifying the changes, issues defects and 'to do' items associated with a particular model element. You can include the maintenance items in the main RTF documentation and HTML produced by Enterprise

Architect. The **RTF Setup** dialog has checkboxes to show or hide element maintenance items.

Access: **View | Other Element Tools | Maintenance (Shortcut: Alt+4)**

Topics:

Topic	Detail	See also
Usage	<p>You can also use the Element Browser window to select and display specific items on the Maintenance window. Click on the  icon in the Maintenance window toolbar to display the Element Browser, open the Maintenance folder and select the required item. In the folder, the 'page' icon contains a C for Change items, D for Defect items, T for Task items, or I for Issue items.</p> <p>The window has two formats. In List mode a detailed list is provided showing several columns of information. In Item mode this list is cut down and instead an editor for the selected item is provided.</p> <p>To toggle between the modes, click on the Show/Hide Properties button in the window toolbar. Item mode displays a single item with others of the same type listed in the left-hand panel. You can also either switch to List mode or select another item from the Element Browser window. List mode displays all items of one type in the selected element; it does not, however, display as much detail on an item as Item mode does.</p> <p>Using the toolbar, you can add or delete items and show or hide the Properties window to enable you to edit each item in the list. Click on the New icon in the window toolbar to add new items. In Item mode, this clears the fields for new data. In List mode, this displays the <item type> details for <element type> <element name> dialog. By clicking on the Auto button in Item mode or on the details dialog, you can apply an automatic naming/numbering nomenclature that you have previously defined.</p> <p>An asterisk on a tab (as for the Defects tab, above) indicates that the tab contains saved information. If the tab has no information or the information has not yet been saved, there is no asterisk, as shown for the Tasks tab.</p> <p>You can also display the maintenance items in a compartment of each appropriate element in a diagram.</p>	<p>Element Browser^[687]</p> <p>Maintenance Item Properties^[1715]</p> <p>Use Auto naming and Auto counters^[628]</p> <p>Show Maintenance Scripts in Diagrams^[1717]</p>

Learn More:

- [Maintenance](#)^[1715]

16.2.2 Maintenance Item Properties

Access: [View | More Element Tools | Maintenance](#)

To create, edit or delete maintenance items, follow the steps below

How To:

Step	Action	See Also
1.	Select the View More Element Tools Maintenance menu option. The Maintenance window displays.	
2.	Open a diagram or the Project Browser and select an element. In Item mode, the oldest maintenance item of each type for that element is shown in the Maintenance window, on the appropriate tab. The other items are listed either in the left hand panel or in List mode.	
3.	Click on the Browse Element icon in the window toolbar. The Element Browser window displays.	
4.	To: <ul style="list-style-type: none"> • Add a new item, select the appropriate tab, click on the New icon in the Maintenance window toolbar and complete the fields as described in the table below • Modify an existing item, select the item from the left-hand list panel or from the <i>Maintenance</i> folder of the Element Browser window and edit the fields as described in the table below • Delete an existing item, select the item from the left-hand list panel or the Maintenance folder of the Element Browser window and click on the Delete icon in the Maintenance window toolbar 	
5.	Click on the Save button in the window toolbar.	

Complete or edit the following fields on the **Maintenance** window

Field	Usage	See Also
Name	Type the name or a short description of the defect.	
Reported by	Select the name of the person who reported the defect.	
Reported	Select the date on which the defect was reported.	
Status	Select the defect status, such as New or Complete.	
Resolved by	Select the name of the person who fixed the defect.	
Resolved	Select the date on which the defect was resolved.	
Priority	Select the priority assigned to resolving the defect.	
Version	Type the version number associated with this fix.	
Description	Type a longer description of the defect. You can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be edited there.	Notes Toolbar ¹⁷⁷⁰

Field	Usage	See Also
History	Enter any notes or references to previous occurrences of this defect. You can format the text using the Notes toolbar at the top of the field. This text is also reflected in the Notes window, but cannot be edited there.	

Notes:

- For information on element-level Defects, Issues, Changes and Tasks, see the *Maintenance* topic. For information on the **Maintenance** window, see the *Maintenance Workspace* topic
- This table describes the fields of the **Defects** tab of the **Maintenance** window. The **Changes**, **Issues** and **Tasks** tabs differ only in minor details

Learn More:

- [Maintenance](#)^[1713]
- [Maintenance Workspace](#)^[1713]

16.2.3 Move or Copy Maintenance Items

When you define an item on the **Defects**, **Changes**, **Issues** or **Tasks** tab of the **Maintenance** window, you might decide that the item either is better suited to another Maintenance category, or forms a good template for items in other categories. Enterprise Architect enables you to move or copy items between categories.

To move or copy a maintenance item, follow the steps below

How To:

Step	Action	See Also
1.	Open the Maintenance window and select the tab that contains the item you want to move or copy.	Maintenance Workspace ^[1713]
2.	Right-click on the required maintenance item. The item context menu displays.	
3.	Click on the appropriate option - Move to or Copy to . A list of maintenance categories displays.	
4.	Click on the category to which to move or copy the item. A confirmatory prompt displays.	
5.	Click on the Yes button to confirm the move or copy.	
6.	Click on the target tab to ensure that the item has been added, and make any required changes.	Maintenance Item Properties ^[1715]

16.2.4 Create Elements From Maintenance Item

A maintenance item identifies a defect, change, issue or task concerning an element. The maintenance item could itself be represented by an element if it has wider implications for the project or identifies - for example - an actor, activity or action that requires further definition.

You can create one or more elements from any maintenance item, using the **Maintenance** window. The new element is connected to the maintenance item's parent element by a Dependency connector. The item itself remains unchanged as a characteristic of its parent element.

To create an element from a maintenance item, follow the steps below:

How To:

Step	Action	See Also
1.	Open the Maintenance window and select the tab that contains the item you want to create the element from.	Maintenance Workspace ^[1713]
2.	Right-click on the required maintenance item. The item context menu displays. select the Create as new Element context menu option. The New Element dialog displays.	Add Elements Directly To a Package ^[627]
3.	In the Name field, type a name for the new element.	
4.	In the Type field, click on the drop-down arrow and select the required element type. For example , you might create an Issue element for a Defect or Issue maintenance item, a Change element for a Change item, or an Action for a Task item. You can, however, create a wide range of other element types should any of these be appropriate, and use the Select Group button to select a profile, MDG Technology or Add-In to create an element specific to that element group.	
5.	If necessary, in the Stereotype field click on the drop-down arrow and select a stereotype to apply to the new element.	
6.	If you want to immediately define the properties of the element, select the Open Properties Dialog on Creation checkbox.	
7.	If you are adding multiple elements in one session, deselect the Close dialog on OK checkbox.	
8.	If you want to add the element to the currently-open diagram, select the Add to Current Diagram checkbox.	
9.	Click on the OK button to create the element.	

The element is added to the **Project Browser** and - if requested - to the current diagram.

16.2.5 Show Maintenance Script in Diagram

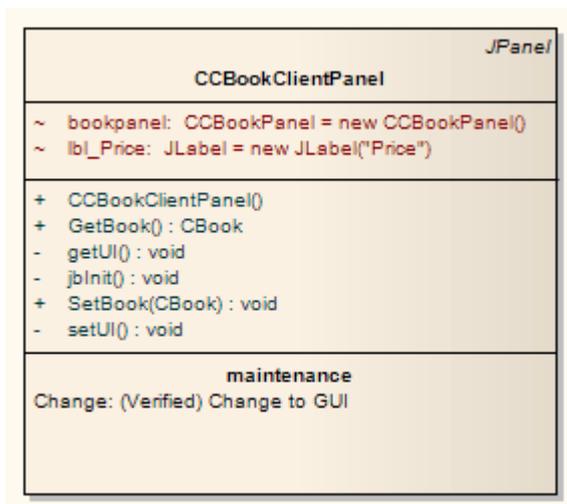
Any element that is capable of displaying a compartment can show **maintenance scripts** in a diagram. To make use of the feature the element must have an attached maintenance item.

To show maintenance compartments in a diagram, follow the steps below

How To:

Step	Action	See Also
1.	Open a diagram containing the element with the attached maintenance items.	
2.	Double-click on the diagram background to display the Diagram Properties dialog. Click on the Elements tab.	
3.	In the Show Compartments panel, select the Maintenance checkbox.	
4.	Click on the OK button to save the setting.	

The maintenance Items now appear as items in the maintenance scripts compartment of the diagram element.

Example:**Learn More:**

- [The Maintenance Workspace](#) ¹⁷¹³

16.3 Changes and Defects

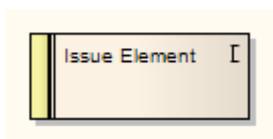


Topics:

Topic	Detail	See also
Change and Defect Elements	<p>Changes and Defects are structured comments that can be used in managing change in a project</p> <p>A Defect element (also known as an Issue element) corresponds to a failure to match the requirements for the current system</p> <p>A Change element corresponds to a change in requirements for the current system</p>	Changes ^[1721] Defects ^[1719]
Using Structured Comments	<p>You can track changes and defects (issues) in an Enterprise Architect model</p> <p>Change and Defect elements can be created in UML diagrams and connected using Realization, Dependency, Aggregation and other relationships to show what model element each affects and how each is resolved</p> <p>You can edit the element properties, and assign people (as Actor elements) to changes and defects</p>	Element Properties ^[1722] Assign People to Defects or Changes ^[1722]

16.3.1 Defects (Issues)

A *Defect* (or *Issue*) element is a structured comment that contains information about defects and issues that relate to the system or model. This corresponds in some sense to a failure to meet defined requirements for the current system. An Issue element looks the same as a Requirement element:



Enterprise Architect enables you to generate and handle issues in much the same way as you can handle and **color code** Requirements. See the **Requirements** topic for more information.

You can link Issues using *Realize* connectors to model elements that are responsible for the defect. You can even structure a hierarchy of Issues using aggregation.

To add an Issue to the model using the Toolbox, follow the steps below

How To:

Step	Action	See Also
1.	Open a Custom diagram.	
2.	From the Custom pages or Common page of the Toolbox, drag the Issue icon onto the diagram.	Custom Group ^[562] Common Group ^[552]
3.	Enter the details as required.	

To add an Issue to the model using the Insert New Element dialog, follow the steps below

How To:

Step	Action	See Also
1.	Right-click on a package in the Project Browser .	
2.	Select the Insert New Element context menu option. The New Element dialog displays.	
3.	In the Type field, click on the drop-down arrow and select Issue .	
4.	In the Name field, type a name for the element.	
5.	Click on the OK button.	

Notes:

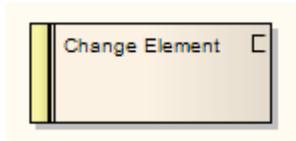
- Issue elements can be created with or without an identifying I in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the **Options** dialog, **Objects** page

Learn More:

- [Color Codes](#)^[1159]
- [Requirements](#)^[1152]
- [Objects](#)^[432]

16.3.2 Changes

A *Change* element is a structured comment that contains information about requested changes to the system/model. This corresponds in some sense to a change in requirements for the current system. A Change element looks the same as a Requirement element:



Enterprise Architect enables you to generate and handle Changes in much the same way as you can handle and [color code](#) ^[1159] Requirements. See the [Requirements](#) ^[1152] topic for more information.

You can connect *Changes* using *Realization* connectors to model elements that implement the Change, and you can structure a hierarchy of changes using Aggregation.

Change elements can be created with or without an identifying **C** in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page.

How To:

To add a Change to the model using the Toolbox, follow the steps below

Step	Action	See Also
1	Open a Custom diagram.	
2	From the Custom pages or Common page of the Toolbox, drag the Change icon onto the diagram.	Custom Group ^[562] Common Group ^[552]
3	Enter the details as required.	

To add a Change to the model using the Insert New Element dialog, follow the steps below

Step	Action	See Also
1	Right-click on a package in the Project Browser .	
2	Select the Insert New Element context menu option. The New Element dialog displays.	
3	In the Type field, click on the drop-down arrow and select Change .	
4	In the Name field, type a name for the element.	
5	Click on the OK button.	

Notes:

- Change elements can be created with or without an identifying **C** in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements**

checkbox on the **Options** dialog, **Objects** page.

Learn More:

- [Color Codes](#) ^[1159]
- [Requirements](#) ^[1152]
- [Objects](#) ^[432]

16.3.3 Element Properties

The **Properties** dialog for Changes and Issues is similar to that used by Requirements. It has a **Properties** tab containing the name of the Issue and relevant management details (such as owner and dates). You can also associate files with the issue and add Tagged Values.

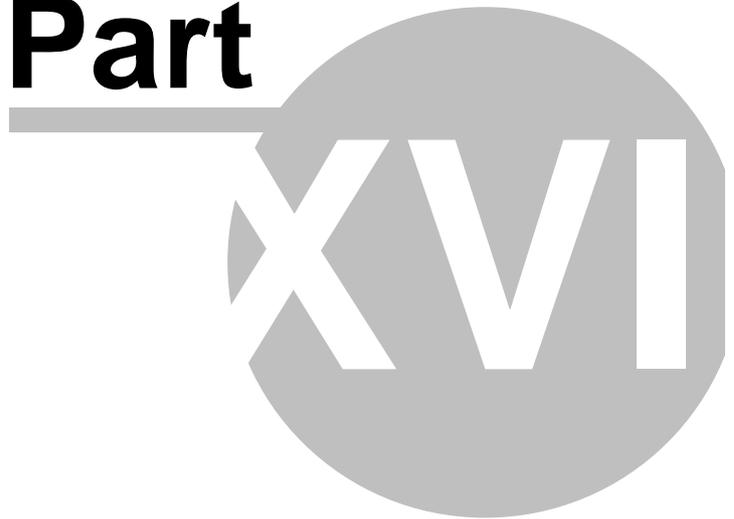
Learn More:

- [Associate Files](#) ^[687]
- [Add Tagged Values](#) ^[763]

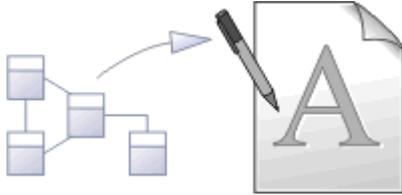
16.3.4 Assign People to Defects or Changes

As an example of how you might use the **Relationship Matrix** to monitor issues or changes, the screen below illustrates staff (actors) being linked through *Realization* connectors to *Issues*. Each highlighted square indicates a responsibility of a staff member to work on or correct a named issue. This same approach can be used for any mix of model elements.

Part



17 Reporting



Model documentation is essential to realizing the full benefit of Enterprise Architect. Enterprise Architect provides a powerful mechanism for generating high quality, customized documentation directly from your model, in either RTF or HTML format.

Topics:

Topic	Detail	See also
Specifying Content	<p>There are many ways to specify the Enterprise Architect content being documented; you can:</p> <ul style="list-style-type: none"> • Document a package and/or its child packages by manually highlighting the package and selecting a documentation control • Specify embedded packages for exclusion if child packages are recursively documented • Link a package to an RTF document template to simplify generating consistent types of documentation (e.g. Use Case Reports) using the Documents feature • Select, group and order packages together in a different manner from the Project Browser by creating 'Virtual Documents', either linked through a master document with headers, footers and contents list, or as separate individual documents 	<p>Exclude Package From Report^[1731]</p> <p>Save As Document^[1793]</p> <p>Virtual Documents^[1774]</p>
RTF Documentation	<p>Rich text reports are documents produced by Enterprise Architect in Rich Text Format (RTF), which can be modified directly with RTF Style templates to alter the look and feel of generated output</p> <p>Using MS Word you can further enhance the separate RTF documents output from the model by connecting and interweaving them into a linked master document with headers, footers and contents list</p> <p>Enterprise Architect has a fully-featured RTF Document Generator that features:</p> <ul style="list-style-type: none"> • Powerful WYSIWYG RTF style template editor support • An easy-to-use document generator • An embedded RTF viewer that enables you to view RTF documents generated by Enterprise Architect within Enterprise Architect 	<p>RTF Documents^[1726]</p> <p>Use MS Word^[1795]</p>
RTF Reports	<p>You can also generate a number of RTF reports on different aspects of your model</p>	<p>Other Documents^[1783]</p>
HTML Documentation	<p>Enterprise Architect provides automated web-based publishing of models, making it simple to explore large models efficiently on-line</p> <p>Enterprise Architect enables the export of an entire model or a single branch of the model to HTML Web pages; you can also</p>	<p>HTML Reports^[1802]</p>

Topic	Detail	See also
	create web style templates to customize the HTML output	

17.1 RTF Documents



Topics:

Topic	Detail	See also
Rich Text Format Documentation	<p>Rich text reports are documents produced by Enterprise Architect in Rich Text Format (RTF), a format common to many word processors</p> <p>You can also maintain complex documents by creating virtual documents in Enterprise Architect, setting up a Master Document (package) element and/or Model Document elements (Class elements of stereotype Model Document) and linking packages into the document, in whatever order or combination is most appropriate to your requirements</p> <p>You can select packages from different areas of the model, arrange them in any order, and edit or delete the packages; the virtual document automatically incorporates the changes each time you generate it</p> <p>Alternatively, as RTF is particularly targeted at Microsoft Word™, it also enables you to link a number of rich text documents into a single master document</p> <p>Typically you create a Word master document, then some Enterprise Architect RTF reports, and you link the reports back into sub-sections of the master document and refresh them as required during project development; in this way the project document becomes an easily-managed and feature-rich work product</p> <p>You can also populate a Word document from specific sections of reports, based on bookmarks - for example, a Word document might have a section for a small part of your component model; using bookmarks you can generate a full component model, and then link into just one section of the report</p> <p>This way you can maintain a complex Word document from parts of Enterprise Architect reports; the RTF Generator performs one pass for one template, but using a Word master document and Enterprise Architect bookmarks enables you to incorporate material from several RTF documents with different formats based on different templates</p> <p>By adding tables of contents, figure tables, sections, and headers and footers, you can manage a complex document with relative ease; simply update the Enterprise Architect RTF reports then refresh the links in MS Word</p> <p>You can have several RTF reports open at the same time, as</p>	<p>Virtual Documents [1774]</p> <p>Using MS Word [1795]</p> <p>RTF Bookmarks [1796]</p> <p>Diagram Tabs [543]</p>

Topic	Detail	See also
	separate tabs in the central view area of the Enterprise Architect work area; you can also close the reports individually or all together, leaving views of other types (such as diagrams or code editors) still open	
The RTF Generator	Enterprise Architect has an enhanced RTF Document Generator that features: <ul style="list-style-type: none"> • Powerful WYSIWYG RTF style template editor support, enabling: <ul style="list-style-type: none"> • Headers and footers • Images • Indexes • Tabular Sections • Nested Sections • All model elements, connectors, diagrams and their properties • Template import and export using XML • Basic templates supplied for customization. • A document generator that: <ul style="list-style-type: none"> • Provides simplified options • Generates complex documents based on RTF templates • Has an embedded RTF viewer that you use to view RTF documents generated in Enterprise Architect directly within Enterprise Architect 	
More Information	A tutorial on using the RTF Generator and creating RTF documentation is provided on the Sparx Systems website	http://www.sparxsystems.com/resources/whitepapers/

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Documents** permission to generate RTF documents

Learn More:

- [Generate RTF Documents](#)^[1727]
- [Generate RTF Documentation Dialog](#)^[1732]
- [RTF Document Options](#)^[1767]
- [RTF Templates Tab](#)^[1734]
- [Custom Language Settings](#)^[1794]
- [Include or Exclude a Package from Report](#)^[1731]
- [List of Available Permissions](#)^[205]

17.1.1 Generate RTF Documents

Creating a Rich Text Format (RTF) document is a simple and flexible process. An RTF document is based on a package or an element in your project (more usually a package). To produce a document, you must select the package or element to report on in the Project Browser, Package Browser, Diagram List or Model Search.

You should also set the diagram properties to determine how the diagrams in the package are set out in the RTF document. When you have prepared and selected your package, use the context menu to open the Generate RTF Documentation dialog and configure the details of your document. The next topic guides you through creating a rich text report.

Topics:

Topic	Detail	See also
Open the Generate RTF Documentation Dialog	<p>Use one of the following methods:</p> <ul style="list-style-type: none"> • Select the Project Documentation Rich Text Format (RTF) Report menu option • In the Project Browser, right-click on the required package and, on the context menu, select the Documentation Rich Text Format (RTF) Report menu option • In the Project Browser or a diagram, click on the required package or element and press (F8) • In a diagram, click on a specific element and select the Element Advanced Rich Text Format (RTF) Report menu option • In the Diagram List, Package Browser or Model Search, select one or more items, right-click and, from the context menu, select either the RTF Report Generate report for each selected object option or the RTF Report Generate one report for all selected option 	Generate RTF Documentation Dialog ^[1732]
Generate RTF Report From Package Browser or Model Search	<p>When you select to create an RTF Report from the Diagram List, Package Browser or Model Search tools, you can generate an element-level report rather than a package-level report, and you have additional flexibility in selecting:</p> <ul style="list-style-type: none"> • The type of element to report on • The specific elements to report on, together or separately, whether in the same package or not <p>For example, you might want to find all elements with test cases, with the intention of reporting on some or possibly all such elements; with the Diagram List or Package Browser, you would identify these elements yourself within the list of all elements in a selected package, but with the Model Search you could specifically identify the elements across a section of the model or across the whole model, as required</p> <p>The search filtering could be for specific test cases; however, the results are by element so if there are test cases outside the range in any element that has a filtered test, these elements are listed as well</p> <p>Having generated the list of elements, you can select individual elements, blocks of elements, or all elements and then (as above) use the context menu to generate a report on all of the elements, or separate reports on each element</p>	Diagram List ^[462] Package Browser ^[456] Model Search ^[475]

How To:

To generate and view an example RTF report right now, follow the steps below:

Ste	Action	See Also

p		
1	Open the <i>EASample</i> project	
2	Open the <i>QA Model</i> package and right-click on the Testing package	
3	Select the Documentation Rich Text Format (RTF) Report context menu option The Generate RTF Documentation dialog displays	
4	In the Output to file field, select a convenient file location in which to hold the generated report	
5	In the Use Template field, click on the drop-down arrow and select (basic template)	
6	Click on the Generate button	
7	When the report has been generated, click on the View button	

Notes:

- Reports can be configured to include all packages within a parent package, or just the top level

Learn More:

- [Diagram Options](#)  ^[1729]

17.1.1.1 Diagram Options

This topic refers to options on the Diagram page of the Diagram Properties dialog, which are used when generating RTF reports for a particular diagram.

Access: **Diagram background context menu | Properties > Diagram page**

Topics:

Topic	Detail	See also
Exclude Image from RTF Documents	Select this checkbox to exclude the image of the current diagram from any RTF reports	
Document Each Contained Element in RTF	Select this checkbox to ensure the RTF report generator includes details of elements that belong in other - external - packages but that are linked into this diagram within the package being reported on; the report includes external elements for each diagram on which you have selected the checkbox On the Generate RTF Documentation dialog, the Include all Diagram Elements in Report checkbox defaults to selected, to include external elements for <i>every</i> diagram covered by the report; therefore, to include external elements only for specific diagrams, deselect this checkbox In either case, you must enable the <i>Package.Diagram.Element</i> and <i>Package.Element</i> sections in your (customized) document generation templates in order to include this external element	Generate RTF Documentation Dialog  ^[1732] Report Elements From External Packages  ^[1730]

Topic	Detail	See also
	information in the report	
Divide Diagram Into Multiple Sections	Select this checkbox to divide each large diagram into separate pages in the RTF document This option is only effective when the Scaled Printing option is set to None on the Print Advanced dialog	Scaled Printing Option ^[602]
Rotate Images	Select this checkbox to rotate each diagram image by 90 degrees in the RTF document; this is only valid for bitmap (.bmp) images	

Learn More:

- [Diagram Tab](#)^[571]

17.1.1.2 Report Elements From External Packages

Elements can be re-used in different diagrams across a model; this can often mean that a diagram contains elements that are held in packages other than the diagram's parent package.

When generating an RTF report on a package, you might want to include elements that are not held in that package, but that are referenced in diagrams within the package. To do this you must:

- Select the appropriate diagram options on the Generate RTF Documentation dialog and/or the Diagram page of the diagram Properties dialog
- Enable the appropriate template sections in the report template, as described in this topic

Access: [Project Browser package context menu | Documentation | Rich Text Format \(RTF\) Report \(F8\)](#)

How To:

To enable the template sections, follow the steps below:

Step	Action	See also
1	Select the Rich Text Format (RTF) Report menu option The Generate RTF Documentation dialog displays	
2	In the Use Template field, click on the drop-down arrow and select the appropriate customized template (not a system-provided one)	
3	Click on the Edit Template button to display the RTF Template Editor	
4	In the Sections panel on the left-hand side of the editor window, select the Package::Element checkbox Selecting the checkbox adds the following set of sections to the Content panel of the template:	

Step	Action	See also
	<pre> package > [right-click-to-insert-Package-field(s)] element > [right-click-to-insert-Element-field(s)] child elements > < child elements < element < package </pre>	
5	<p>Delete the <i>[right-click-to-insert-Element-field(s)]</i> text to leave a blank area</p> <p>Right-click and select the Insert Field Name context menu option, this inserts an Element.Name field in the template</p>	
6	<p>In the Sections panel, select the Package::Diagram::Element checkbox</p> <p>Selecting the checkbox adds the following set of sections to the Content panel of the template:</p> <pre> package->¶ [right-click-to-insert-Package-field(s)]¶ diagram->¶ [right-click-to-insert-Diagram-field(s)]¶ element->¶ [right-click-to-insert-Element-field(s)]¶ <-element¶ <-diagram¶ <-package¶ </pre>	
7	<p>To report on the linked elements in:</p> <ul style="list-style-type: none"> the same style as defined in the Package::Elements section, delete the <i>[right-click-to-insert-Element-field(s)]</i> text to leave a blank area a different style to the elements within the selected package, set up the style as appropriate <p>If using the Legacy RTF generator, the <i>Element</i> checkbox is automatically checked in your customized template, when you select the Document each contained element in RTF checkbox</p> <p>Package::Diagram::Element, if left blank, replicates the format of Package::Element including sub-element sections (for example, Package::Element::Scenario); Package::Diagram::Element does not have an option to add these sections</p>	

Learn more:

- [Diagram Options](#) ^[1729]
- [Learning Center](#) > Reporting | RTF Templates | Reporting Elements from External Packages

17.1.1.3 Exclude Package from Report

You can exclude a particular package from any RTF or HTML report by marking it for exclusion.

Access: [Project Browser package context menu](#) | [Documentation](#) | [Generated Report Options](#)

How To:

To mark a package for exclusion from reporting, follow the steps below:

Step	Action	See Also
1	Select the Generated Report Options menu option The Generated Report Options dialog displays	
2	Select the Exclude Package from Generated Reports radio button	
3	Click on the OK button	

Notes:

- By default, packages are included in any generated reports
- When you exclude a package from reports, all of the selected package's subpackages are also excluded

17.1.1.4 Generate RTF Documentation Dialog

The Generate RTF Documentation dialog enables you to set the exact contents and look and feel of your report.

Each tab of the dialog offers a number of RTF document generation options, as described in the following sections:

- Generate Tab (see below)
- RTF Templates
- Document Options
- Advanced Options
- Element Filters
- Other Filters
- Project Constants
- Word Substitution
- Language Substitution (Codepage)

When you have worked through these tabs and set the options you require, return to the Generate tab and click on the **Generate** button to produce your report.

Reference:

Option	Action	See also
Model Document	Confirm the name of the element selected from the Project Browser, Diagram List or Model Search	Virtual Document ^[1774]
Root Element Root Package	If this is the specially-created model document element for a Virtual Document, the field is Model Document Otherwise, this field identifies the selected element of the hierarchy to be reported on; that is, the Root Element or Root Package	
Output to File	Type or select the location and filename for the generated documentation The (...) (Browse) button enables you to navigate to the location	

Option	Action	See also
Use Template	<p>Type or select the name of the RTF template to apply to document generation</p> <p>You can select either a standard template (enclosed in parentheses) or a user-generated template</p> <p>The standard templates include the following:</p> <ul style="list-style-type: none"> • (basic template) • (diagrams template) • (maintenance template) • (requirements template) 	
Use Language Substitutions	<p>Switch custom language word substitutions on</p> <p>Deselect to switch custom language word substitutions off</p>	Word substitutions 1773
View Document On Completion	<p>Open the document as soon as it has been generated</p>	
Use Internal Viewer	<p>Enable the View button to launch the generated RTF Documentation in the Enterprise Architect internal viewer</p> <p>Deselect to enable the View button to launch the generated RTF Documentation in the MS Windows default RTF file viewer</p>	Document options 1767
Include all Diagram Elements in Report	<p>Include elements in the report from external packages that are referenced from a diagram, for every diagram covered by the report; this defaults to selected</p> <p>The <i>Package.Diagram.Element</i> and <i>Package.Element</i> checkboxes must be selected in the current template</p> <p>If external elements are to be included only for specific diagrams, deselect this checkbox and - only for the diagrams for which you want to include external elements in the report - for each diagram select the Document each contained element in RTF checkbox in the diagram properties</p> <p>When both options are deselected, or when neither of the <i>Package.Diagram.Element</i> or <i>Package.Element</i> checkboxes are selected in the template, only elements in the current package are documented</p>	Diagram properties 1729
Optimize for Open Office	<p>Generate reports where diagrams are adjusted for clean rendering in Oracle Open Office</p> <p>The option also overwrites the value fields defined in a document section with the actual value text.; if you open a generated report in Word, you can format the text but you cannot display the original field code</p> <p>If the checkbox is <i>deselected</i>, diagrams displayed in Open Office are less distinct - value fields remain and are populated with the appropriate values; when you display a generated report in Word, you can right-click on the field and toggle between the value text and the field code</p> <p>If you use Open Office as your text editor, you must select this checkbox to show the field values; Open Office defaults to showing the field codes only, and you cannot toggle to the field values</p>	
Generate	<p>Generate the document (after you have set all the options you</p>	

Option	Action	See also
	require, on all tabs of the dialog)	
View	Launch the generated RTF Documentation in the MS Windows default RTF file viewer, or in the Enterprise Architect internal viewer if you have selected the Use Internal Viewer checkbox	
Edit Template	Edit the currently-named template using the RTF Style Template Editor You can only edit <i>user-defined</i> templates, not the standard templates provided with Enterprise Architect; standard template names are enclosed in parentheses	RTF Style Template Editor ^[1736]
Resource Document	Save the current options as a document definition	Document definition ^[1766]
Abort	Cancel report generation	

Notes:

- For an introduction to generating RTF documentation, see the *RTF Documents* topic
- If you have never selected the **Optimize for Open Office** checkbox, the document generator prompts you to confirm which document style you want to adopt - Microsoft Office or Oracle Open Office; once you have made a selection in this prompt, the document generator always acts on the status of the checkbox and does not display the prompt again

Learn More:

- [RTF Documents](#) ^[1726]
- [RTF Templates](#) ^[1734]
- [Document Options](#) ^[1767]
- [Element Filters](#) ^[1770]
- [Other Filters](#) ^[1771]
- [Project Constants](#) ^[1772]
- [Word Substitution](#) ^[1773]
- [Language Substitution \(Codepage\)](#) ^[1773]

17.1.1.5 RTF Templates Tab

The Templates tab of the Generate RTF Documentation dialog enables you to create, edit and delete your own RTF style templates. You can also import RTF templates saved as XML files.

Topics:

Topic	Detail	See also
Abstract	In versions of Enterprise Architect later than 7.1, a <i>Normal.rtf</i> template is provided as a system template, as an external file stored in: <code>%APPDATA%\Sparx Systems\EA\RTF Templates</code> This provides user-editable defaults of styles, numbering and other base formats; any styles modified in the <i>Normal.rtf</i> file reflect in newly created templates - one useful style is the default list numbering, <i>MasterList</i>	File and Print Options ^[1745]

Topic	Detail	See also
	<p>To edit the Normal.rtf template, use the RTF Style Template Editor to create a new template called, for example, <i>Normal</i>, and use the File Import menu option to import the Normal.rtf file into the Normal template; modify the Normal template as required, but ensure there is no text when you save it, just style definitions</p> <p>Export the modified template (File Export) back into the Normal.rtf file in the <i>RTFTemplates</i> folder</p> <p>A related feature in the RTF Style Template Editor is the File Update Styles menu option, which enables you to update existing templates to reflect any changes to <i>Normal.rtf</i></p>	

Reference:

The Templates tab has the following functions:

Field	Field	See also
Delete a template	Click on the template name and click on the Delete button	
Create a new template	<p>Click on the New button</p> <p>The New Document Template dialog displays, on which you specify the template name and the name of any existing template to act as the base for the new template</p> <p>To make it easier to get up and running, Enterprise Architect provides a basic template with default settings on which you can base new templates.</p> <p>Modify the template as required, using the RTF Style Template Editor.</p>	
Open the RTF Style Template Editor	<p>Click on the template name and click on the Edit button</p> <p>The Template: <Template name> screen displays, presenting the facilities of the RTF Style Template Editor.</p>	

To Import RTF Templates saved to XML files using the: **Project | Model Import/Export | Export Reference Data** menu option, follow the steps below

How To:

Step	Action	See Also
1.	Click on the Import From Reference File button.	
2.	On the Import Reference Data dialog, click on the Select File button and browse for and select the required file.	
3.	In the Select Datasets to Import panel, click on the required datasets.	
4.	Click on the Import button to import the template.	

The imported template displays in the list on the Templates tab of the Generate RTF Documentation dialog.

Notes:

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is switched on, you must have **Configure Resources** access permission to create RTF templates
- In the Normal.rtf template, you must not edit the *SSbookmark* style. This defines the section styles and must be preserved
- There are two methods of exporting and importing RTF templates out of and into models:
- If the template is in a batch file, export it using the **Project | Model Import/Export | Export Reference Data** menu option and import it using the **Templates** tab, as above
- If the template is a one-off copy, use the RTF Template Editor **File** menu options, **Export** and **Import**

Learn More:

- [RTF Style Template Editor](#)^[1736]
- [File and Print Options](#)^[1745]
- [Update Styles](#)^[1745]
- [Permission List](#)^[205]

17.1.1.5.1 RTF Style Template Editor

The **RTF Style Editor** enables you to create and edit custom RTF templates to define output RTF documentation associated with various sections of the *RTF Report* facility in Enterprise Architect. You typically use this facility to customize the look and feel of a report for your company or client.

Access: **View | More Project Tools | Project Resources > (one of the following folders) Templates, Document Templates, System or Model : View/Modify**

Topics:

Topic	Detail	See also
Usage	<p>You select particular model components and specify, from the component type, the fields to include in the generated document. You can define formatting styles in the RTF Style Template Editor, and add a range of items such as tables of contents or headers to the document.</p> <p>For information regarding specific commands to alter the format of the RTF documentation, see the entries under the RTF Style Template Editor Options topic.</p>	<p>Selecting Components for Recording^[1737]</p> <p>RTF Style Template Editor Options^[1743]</p>

Notes:

- You can transport these RTF templates between models, using the **Export Reference Data** and **Import Reference Data** options on the **Project | Model Import/Export** menu
- You can also access this in alternative ways, by:
 - Selecting to edit the current template on the **Generate** tab of the **Generate RTF Documentation** dialog
 - Selecting to edit a style template on the **Templates** tab of the **Generate RTF Documentation** dialog

Learn More:

- [Resources](#) ^[789]
- [Export Reference Data](#) ^[237]
- [Import Reference Data](#) ^[239]

17.1.1.5.2 Select Components for Reporting**How To:**

To select model components to be documented in the report, using the RTF Style Template Editor, follow the steps below:

Step	Action	See Also
1	Expand the Sections tree on the <template name> screen.	
2	Select the checkbox next to the component name; the component name is then displayed as a section tag in the Content panel. More specific guidance is provided for selecting the following components: <ul style="list-style-type: none"> • Linked Documents and Document Artifact Contents • Tabular Sections • Child Sections • Constraint and Scenario Sections 	Linked Documents and Document Artifact Contents ^[1737] Tabular Sections ^[1738] Child Sections ^[1740] Constraint and Scenario Sections ^[1741]
3	Add content to each report component. The position of the section tags within the Sections tree determines the position of the model component in the Content panel. For encapsulated components, selecting a child component automatically selects the parent also.	Add content ^[1742]

How To:

To move a model component to a different position in the documentation template, follow the steps below

Step	Action	See Also
1	Select the component in the Sections panel.	
2	Click on  and  to move the component up and down the Sections panel.	

Topics:

Topic	Detail	See also
Linked Documents and Document	Linked documents and documents created via the Document Artifact element are rendered into RTF Documentation by selecting the Linked Document checkbox in the RTF Style Template Editor.	Linked Documents ^[729] Creating Document

Topic	Detail	See also
Artifact Contents	<p>The Linked Document checkbox is within the Element hierarchy, towards the end. Remember that checkboxes can be moved up and down the hierarchy to position information in the generated document as you require. In some templates, the Linked Document checkbox is only available as a child of the External Requirements checkbox.</p> <p>The linked document is rendered into the RTF documentation at:</p> <p><i>linked document ></i></p> <p><i><linked document</i></p>	Artifacts ^[730]

Notes:

- If the Package.Diagram.Element checkboxes are selected, you must select either the **Document each contained element in RTF** checkbox in the **Diagram Properties** dialog for at least one included diagram, or the Include all **diagram elements in report** checkbox on the **Generate RTF Documentation** dialog

Learn More:

- [Diagram Options](#) ^[1729]
- [Generate RTF Documentation](#) ^[1732]

17.1.1.5.3 Tabular Sections

The **RTF Style Template Editor** supports rendering a document section as a table. A tabular section is defined as a table containing any number of columns, but with only *two* rows:

- The first row is used to describe the headings of the columns, which you type in and format yourself
- The second row defines the output, which you specify by right-clicking in each cell and selecting the output type from the field list; the output is then generated iteratively for every occurrence of the section in question

Topics:

Topic	Detail	See also
Example Tabular Section	<p>In the following example, the Model> Glossary> section is defined as a tabular section:</p> <pre>model > Model Glossary glossary > Term Type Meaning [ModelGlossary.Term] [ModelGlossary.Type] [ModelGlossary.Meaning] < glossary < model</pre> <p>This renders the following document output:</p>	

Topic	Detail	See also									
	<h3>Model Glossary</h3> <table border="1"> <thead> <tr> <th>Term</th> <th>Type</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>Accounting Periods</td> <td>Business</td> <td>A defined period of time whereby performance reports may be extracted. (normally 4 week periods).</td> </tr> <tr> <td>Association</td> <td>Technical</td> <td>A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link.</td> </tr> </tbody> </table>	Term	Type	Meaning	Accounting Periods	Business	A defined period of time whereby performance reports may be extracted. (normally 4 week periods).	Association	Technical	A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link.	
Term	Type	Meaning									
Accounting Periods	Business	A defined period of time whereby performance reports may be extracted. (normally 4 week periods).									
Association	Technical	A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link.									

Notes:

- Under some circumstances, a table might repeat the header row rather than the output row; if this occurs, create another row in the table between the header row and the output row, and leave this blank
- If you type a carriage return between the end of the table and the section terminator, the table you generate has a line space between the rows. For example:

```

package">
{
}
element">


| Element Name | Author         |
|--------------|----------------|
| Element Name | Element Author |


</element>
</package>
    
```

← Carriage Return

This generates the following table:

	Element Name	Author
	Choose Recipient	John Redfern

Line Space

	Manage Contacts	Walter Frederick
--	-----------------	------------------

To avoid this, ensure that there is no carriage return between the end of the table and the section terminator, as follows:

```

package">
{
}
element">


| Element Name | Author         |
|--------------|----------------|
| Element Name | Element Author |


</element>
</package>
    
```

← No Carriage Return

This generates a table with no space between the rows, as follows:

	Element Name	Author
	Choose Recipient	John Redfern
	Manage Contacts	Walter Frederick

17.1.1.5.4 Child Sections

Child sections can be rendered in RTF documentation using one of the following two methods:

- Render model components directly into the RTF as defined by the section's content and fields.
- Render indirectly to the RTF by using a parent section to describe the content.

The second option occurs as a result of creating a section that has a placeholder section tag (that is, no content within the tags). This method is used to create recursive documentation of child packages.

Topics:

Topic	Detail	See also
Example: Rendered Sub-section	<p>This example shows a template with content between the <i>Child Element</i> tags. In this example child elements of the parent are rendered using the <i>Child Elements</i> section because it contains valid content and fields.</p> <pre> element> {Element.Name} Type: → → {Element.Type}..{Element.BaseClasses} Status: → → {Element.Status}..Version: {Element.Version}..Phase: {Element.Phase} Package: → → {Element.ParentPackage}..Keywords: {Element.Tag} Detail: → → Created on: {Element.DateCreatedShort} {Element.Notes} child element> → Child: {Element.Name} <child element <element </pre>	
Example: Non-rendered Sub-section	<p>This example shows a template with no content between the child element tags. In this example, child elements of the parent are rendered using the element section because the child element section is empty. The child element section is used as a placeholder.</p> <pre> element> {Element.Name} Type: → → {Element.Type}..{Element.BaseClasses} Status: → → {Element.Status}..Version: {Element.Version}..Phase: {Element.Phase} Package: → → {Element.ParentPackage}..Keywords: {Element.Tag} Detail: → → Created on: {Element.DateCreatedShort} {Element.Notes} child element> <child element <element </pre>	
Child Document Sections and Their Corresponding Parent Sections	Child Section	Section rendered when us
	Package->Child Package	Package
	Package->Element->Child Element	Package->Element
	Package->Element->Diagram	Package->Diagram

Topic	Detail	See also						
	<table border="1"> <thead> <tr> <th>Child Section</th> <th>Section rendered when us</th> </tr> </thead> <tbody> <tr> <td>Package->Diagram->Element</td> <td>Package->Element</td> </tr> <tr> <td>Package->Diagram->Connector</td> <td>Package->Element->Conne</td> </tr> </tbody> </table>	Child Section	Section rendered when us	Package->Diagram->Element	Package->Element	Package->Diagram->Connector	Package->Element->Conne	
Child Section	Section rendered when us							
Package->Diagram->Element	Package->Element							
Package->Diagram->Connector	Package->Element->Conne							

Notes:

- In principle, it is better to leave Child sections blank to replicate their parent sections. Child sections do not contain the same sub-section detail as their parents; for example: Element::Child Element does not contain sub-sections such as Scenario or Attribute, so where Child Element sections are populated, these sub-sections are not rendered. An exception to this is cases where sub-sections are not required, but different formatting of the section fields is preferred

17.1.1.5.5 Constraint and Scenario Sections

The template **content options** enable you to include sections in your reports for:

- constraints* on Package Elements, Elements, Connectors and Attributes, and
- scenarios* for Package Elements and Elements.

There are additional options in these types of section that enable you to determine what types of constraint or scenario are included in your reports, as described below.

Topics:

Topic	Detail	See also
Constraints	<p>The following constraint options are available for selection, as checkboxes in the Sections panel of the RTF Style Template Editor:</p> <ul style="list-style-type: none"> Pre-Constraint - select this checkbox to include all constraints of the type 'pre-condition' in this section of the report Post-Constraint - select this checkbox to include all constraints of the type 'post-condition' in this section of the report Constraint - select this checkbox to include all constraints that have not been generated in the Pre-Constraint and Post-Constraint sections of the report 	
Scenarios	<p>The following scenario options are available for selection, as checkboxes in the Sections panel of the RTF Style Template Editor:</p> <ul style="list-style-type: none"> Element > Scenario - select this checkbox to include all scenarios in this section of the report; if any of the following sections are also selected, the report includes all scenarios that are not exception paths Element > Scenario > Exception - select this checkbox to include all exceptions for each scenario Element > Scenario > Structured Scenarios - select this checkbox to include all scenario steps in the scenario sections of the report Element > Scenario > Structured Scenarios > Exception - select this checkbox to include the steps for each exception path in the scenario sections of the report 	

Learn More:

- [Selecting Components for Reporting](#)^[1737]

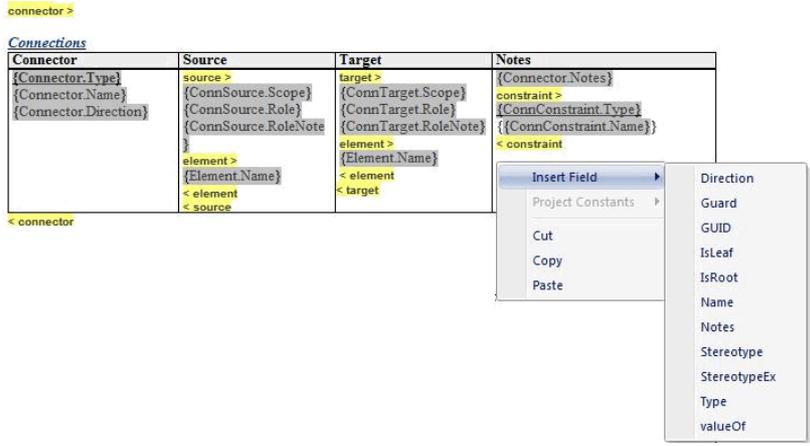
17.1.1.5.6 Add Content

The **RTF Style Template Editor** uses pairs of tags to define the layout of the documentation content. To insert a pair of model component tags into the **Content** section of the Editor, use the **Sections** panel of the **RTF Style Template Editor**.

The beginning of a model component is represented by a yellow highlighted `<sectionname >` tag; the end of the model component is represented by `<sectionname`.

To add model component content, follow the step below

How To:

Step	Action	See Also																																				
1	<p>Right-click in the area between the opening and closing tags. This displays a context menu that enables you to:</p> <ul style="list-style-type: none"> • perform simple text editing (cut, copy and paste) • select from a list of defined project constants to insert at the cursor position; if no project constants have been defined, the menu option is not available • select from a context-sensitive list of fields to add to this component section of the RTF Documentation; any additional information entered between the tags is also included in the generated RTF Documentation. <p>For example, in the <code><connector ></code> section shown below, the user has right-clicked in the fourth column of the table, underneath the <code><constraint</code> tag. They can then select to add another field to the Notes on each connector listed in the table.</p>  <p>The screenshot shows a table with the following structure:</p> <table border="1"> <thead> <tr> <th>Connector</th> <th>Source</th> <th>Target</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>{Connector.Type}</td> <td>source ></td> <td>target ></td> <td>{Connector.Notes}</td> </tr> <tr> <td>{Connector.Name}</td> <td>{ConnSource.Scope}</td> <td>{ConnTarget.Scope}</td> <td>constraint ></td> </tr> <tr> <td>{Connector.Direction}</td> <td>{ConnSource.Role}</td> <td>{ConnTarget.Role}</td> <td>{ConnConstraint.Type}</td> </tr> <tr> <td></td> <td>{ConnSource.RoleNote}</td> <td>{ConnTarget.RoleNote}</td> <td>{ConnConstraint.Name}</td> </tr> <tr> <td></td> <td>element ></td> <td>element ></td> <td>< constraint</td> </tr> <tr> <td></td> <td>{Element.Name}</td> <td>{Element.Name}</td> <td></td> </tr> <tr> <td></td> <td>< element</td> <td>< element</td> <td></td> </tr> <tr> <td></td> <td>< source</td> <td>< target</td> <td></td> </tr> </tbody> </table> <p>The context menu is open over the <code>< constraint</code> tag in the Notes column, showing options like 'Insert Field', 'Project Constants', 'Cut', 'Copy', and 'Paste'. The 'Insert Field' sub-menu is open, showing a list of fields: Direction, Guard, GUID, IsLeaf, IsRoot, Name, Notes, Stereotype, StereotypeEx, Type, and valueOf.</p>	Connector	Source	Target	Notes	{Connector.Type}	source >	target >	{Connector.Notes}	{Connector.Name}	{ConnSource.Scope}	{ConnTarget.Scope}	constraint >	{Connector.Direction}	{ConnSource.Role}	{ConnTarget.Role}	{ConnConstraint.Type}		{ConnSource.RoleNote}	{ConnTarget.RoleNote}	{ConnConstraint.Name}		element >	element >	< constraint		{Element.Name}	{Element.Name}			< element	< element			< source	< target		Project Constants ^[1772]
Connector	Source	Target	Notes																																			
{Connector.Type}	source >	target >	{Connector.Notes}																																			
{Connector.Name}	{ConnSource.Scope}	{ConnTarget.Scope}	constraint >																																			
{Connector.Direction}	{ConnSource.Role}	{ConnTarget.Role}	{ConnConstraint.Type}																																			
	{ConnSource.RoleNote}	{ConnTarget.RoleNote}	{ConnConstraint.Name}																																			
	element >	element >	< constraint																																			
	{Element.Name}	{Element.Name}																																				
	< element	< element																																				
	< source	< target																																				

Topics:

Topic	Detail	See also
The valueOf Field	<p>For certain sections, you can add a field to capture a special characteristic of a model component, as defined by a specific Tagged Value. This is the valueOf field, shown in the list for a connector section in the above graphic. The sections that provide the valueOf field are:</p> <ul style="list-style-type: none"> • Package • Element • Connector • Attribute • Operation • External Requirement <p>When you select the valueOf field from the context menu, the template editor prompts you to specify the tag (Tagged Value) from which to extract the value for the report output. This tag should be one of the tags associated with the model component, such as <code>ConnectorAltName</code> for a connector. When you provide the tag name, the template editor adds the field at the cursor position, in the format:</p> <pre>{Connector.valueOf(tagname)}</pre> <p>for example: <code>{Connector.valueOf(ConnectorAltName)}</code></p> <p>For clarity, you could type some lead-in text or the meaning of the Tagged Value immediately preceding the value field; for example:</p> <p>Alternative Name: <code>{Connector.valueOf(ConnectorAltName)}</code></p>	

Notes:

- If you select a field with short date format (such as *Pkg.DateCreatedShort*, *Diagram.DateModifiedShort* or *Element.DateCreatedShort*) the format is actually drawn from the MS Windows settings. To use a different short date format, click on the **Start** icon on the Windows desktop and select the **Control Panel | Regional and Language Options | Customize** option

Learn More:

- [Select Components for Reporting](#)^[1737]

17.1.1.5.7 RTF Style Template Editor Options

The following topics provide assistance on using the **RTF Style Template Editor**.

Topic	Link
Scroll Through Text	Scroll Through Text ^[1744]
File and Print Options	File and Print Options ^[1745]
Cut and Paste Options	Cut and Paste Options ^[1746]
View Options	View Options ^[1747]
Image and Object Inserts	Image and Object Inserts ^[1749]
Character Formatting	Character Formatting ^[1750]
Paragraph Formatting	Paragraph Formatting ^[1751]

Topic	Link
Tab Support	Tab Support ^[1753]
Page Breaks and Repagination	Page Breaks and Repagination ^[1754]
Headers and Footers	Headers and Footers ^[1754]
Hyperlinks and Bookmarks	Hyperlinks and Bookmarks ^[1755]
Table Commands	Table Commands ^[1756]
Sections and Columns	Sections and Columns ^[1759]
Stylesheets and Table of Contents	Stylesheets and Table of Contents ^[1759]
User-Defined Section Numbering	User-Defined Section Numbering ^[1761]
Frame and Drawing Objects	Frame and Drawing Objects ^[1763]
Search/Replace Commands	Search/Replace Commands ^[1764]

Notes:

- Throughout your template editing, be aware that:
 - To undo one or more immediately previous edits, press (**Ctrl+Z**), or select the **Edit | Undo** menu option. You can still undo a change even after you have saved the change
 - To redo one or more immediately previous undone edits, press (**Ctrl+Y**), or select the **Edit | Redo** menu option

17.1.1.5.7.1 Scroll Through Text

Scroll Using	Options
Keyboard Keys	Press <ul style="list-style-type: none"> (↑), (↓), (←) or (→) to scroll up or down a line, or left or right one character (Home) to move to the beginning of the current line (End) to move to the end of the current line (Ctrl+Page Up) to move to the beginning of a file (Ctrl+Page Down) to move to the end of a file (Page Up) to display the previous page (Page Down) to display the next page (Ctrl+←) to move to the previous word (Ctrl+→) to move to the next word (Ctrl+↑) to move to the first column of the current line (if not already on the first column) or the first column of the previous line (Ctrl+↓) to move to the first column of the next line (F10) (or select the Other Jump menu option), type in a line number to jump to, and click on the OK button
Mouse	Click on the vertical and horizontal scroll bar to perform various scrolling functions. These functions are available only if the horizontal or the vertical bar has been enabled by the startup parameters. Vertical Scroll Bar: Click on the arrows on either end to scroll the screen up or down by one line.

Scroll Using	Options
	<p>Click above the elevator to scroll the screen up by one page. Similarly, click below the elevator to scroll the screen down by one page.</p> <p>You can also drag the elevator to any position in the bar. As the elevator is dragged, the editor scrolls the screen up or down accordingly.</p> <p>Horizontal Scroll Bar: Click on the arrows on either end to scroll the screen left or right by one column. Click on either side of the elevator to scroll the screen left or right by 1/2 screen.</p> <p>You can also drag the elevator to any position in the bar. As the elevator is dragged, the editor scrolls the screen left or right accordingly</p>

17.1.1.5.7.2 File and Print Options

Command	Action	Shortcut	See also
New	<p>Clear an existing template from the edit window and start an empty, unnamed template.</p> <p>The editor prompts you to save any modification to the previous template.</p>		
Revert	Revert to the previously-saved copy of the template.		
Save	<p>Save the text to the current file name.</p> <p>If a file is not yet specified, the editor prompts you for a template name.</p>	Ctrl + S	
Save As	Similar to Save File , but you specify a new template name for saving the template.	Ctrl + Shift + S	
Import	<p>Import an existing RTF document into the Template Editor, so as to insert model elements from that document into the template.</p> <p>This option is useful when creating templates from a predefined document with a particular 'look and feel'.</p>		Import an existing RTF document ^[1765]
Export	<p>Save the template as an RTF document rather than as a template.</p> <p>This can be useful for saving the template for other models.</p>		
Update Styles	<p>Imports the styles from <i>Normal.rtf</i>, found in the RTF Templates directory.</p> <p>This option is useful when creating a Master Document / Sub Documents that require consistent user-defined styles across multiple templates (including such things as numbering formats).</p>		
Document Options	Display the Document Options dialog which enables you to set the filter and order the elements.		Document Options ^[1767]

Command	Action	Shortcut	See also
Page Layout	Specify the page layout, before selecting the Print option. You can specify the margins (left, right, top and bottom) in inches.		
Printer Setup	Invoke a printer-specific dialog for the default printer (the default printer selection is made from the Windows Control panel). You select the parameters from a set of printer-specific options. These options include page size, page orientation, resolution and fonts.	Ctrl + Shift + P	
Print	<p>Print the contents of the current file. The editor displays a dialog where you can select the scope of the printing. You can also choose to print only a selected part of the file.</p> <p>To print a block of text, highlight the required text before invoking the Print function. This command prints a highlighted:</p> <ul style="list-style-type: none"> • Line block • Character block <p>The Print function prints on a default printer selected from the Windows Control panel. You can alter the printer setup or page layout prior to invoking the Print function.</p>	Ctrl + P	
Print Preview	<p>Preview the document before printing. The editor displays up to two pages at a time. You can scroll to a different page using (Page Up) , (Page Down) or the scroll bar.</p> <p>By default the preview rectangle is sized to fit the current window. However, you can use the zoom option to enlarge or shrink the preview rectangle as required.</p> <p>Click on the Edit button or the File Print Preview menu option again to return to editing mode.</p>		
Close	Close the Template Editor. The editor prompts you to save any unsaved information.		

17.1.1.5.7.3 Cut and Paste Options

To	Action	See also
Highlight a word	Double-click on the word.	
Highlight a line	Move the cursor onto the line and press (F8) .	
Select all file content	Press (Ctrl+A) or select the Edit Select All menu option.	
Copy a block	<p>Highlight the lines of text to be copied and press (Ctrl+C) , or select the Edit Copy menu option.</p> <p>Move the cursor to the point at which to insert the text and press (Ctrl+V) or select the Edit Paste menu option.</p>	

To	Action	See also
Move a block	Highlight the lines of text to be moved and press (Ctrl+X), or select the Edit Cut menu option. The selected text is removed from the page. Move the cursor to the point at which to insert the text and press (Ctrl+V) or select the Edit Paste menu option.	
Delete a block	Highlight the lines of text to be deleted and press (Delete)	
Delete a line	Press (Shift+F9) to delete the current line. The remaining lines close up.	
Paste special objects	Select the Edit Paste Special menu option. The Paste Special dialog displays, listing the appropriate data type formats for pasting the copied object, as listed below. Click on the Paste button to embed the data into your application, or click on the Paste Link button to create a link to the original file. <u>Native Object Format</u> If available, this is the first format in the list box. You can edit data in this format using the original application, by double-clicking the object. <u>Formatted Text</u> A text format. This option offers the most suitable format if the data is pasted from another text output application, as the font and formatting attributes are reproduced accurately. <u>Unformatted Text</u> Another text format. This option pastes the text without retaining the formatting information. <u>Picture Format</u> The data is available in Picture format. You can later edit the object, by double-clicking on it and invoking the Microsoft MS Draw application. This format is preferred over the Bitmap and the Device Independent Bitmap formats. <u>Device Independent Bitmap and Regular Bitmap formats</u> The data is available in bitmap formats. You can later edit the object, by double-clicking on it and invoking the Microsoft MS Draw application. The editor converts these formats into the Picture format before calling the drawing application.	

17.1.1.5.7.4 View Options

Menu Option	Action	See also
Page Mode	Turn Page Mode on (the equivalent of <i>Print View</i> in Word) or off (the equivalent of <i>Normal View</i> in Word). In Page Mode , the editor displays one page at a time. It is most useful for documents containing multiple columns, as the	

Menu Option	Action	See also
	columns are displayed side by side.	
Fitted View	Turn Fitted View on or off. This is a special case of Page Mode , in which the text wraps to the window width and the soft page breaks are not displayed. If you select Fitted View , Page Mode is automatically selected too. If you deselect Page Mode , Fitted View is automatically deselected too.	
Ruler	Display or hide the ruler at the top of the page. The ruler shows tab stops and paragraph indentation marks; it can also be used to create or delete tab stops.	
Tool Bar	Display or hide the tool bar above the ruler. The tool bar provides a convenient method of selecting fonts, point sizes, character styles and paragraph properties. The tool bar also shows the current selection for font, point size and character styles.	
Status Ribbon	Display or hide the status ribbon at the bottom of the editing panel. The status ribbon displays the current page number, line number, column number and row number. It also indicates the current insert/overtyping mode.	
Paragraph Marker	Display or hide the paragraph marker (an inverted 'P') at the end of each paragraph. This option is useful when working with lines with many different heights.	
Hidden Text	Show or hide 'hidden' text. Text formatted with the hidden attribute is shown with a dotted underline. When the option is turned off, the hidden text is not visible.	Character Formatting Options <small>[1750]</small>
Field Names	Insert, show and hide field names. As you develop your RTF document, you right-click on sections to insert field markers. You cannot insert these markers unless you have selected the Field Names option. When you deselect the option, existing field names are obscured.	
Page Header/ Footer	Display or hide the text of page headers and footers. If Page Mode is not selected, this option turns Page Mode on. When Page Mode is selected, you cannot edit the header or footer text unless you also select the Edit Edit Page Header/Footer menu option. When Page Mode is deselected, you can see and edit page headers and footers at the start of the document.	Headers and Footers <small>[1754]</small>
Page Border	Display or hide a page outline in Page Mode . When Page Mode is deselected, this option is not available. When Page Border is selected, the document contents are shown within a page outline. When Page Border is deselected,	

Menu Option	Action	See also
	the document contents are formatted within the boundaries of the editing screen.	
Zoom	Shrink or enlarge the display of the document text, by selecting the appropriate percentage enlargement. The editor supports zoom percentages from 25 to 200.	

17.1.1.5.7.5 Image and Object Inserts

To	Action	See also
Embed a picture in the document	Position the cursor at the point at which to insert the picture bitmap or Windows metafile, and either: <ul style="list-style-type: none"> • Select the Insert Embed Picture menu option, or • Press (Alt+F8) . A browser dialog displays, through which you select the picture to embed in the document. The picture displays at the current cursor location. The embedded picture is saved within the document.	
Link a picture file to the document	Position the cursor at the point at which to link the picture bitmap or Windows metafile, and select the Insert Link Picture menu option. A browser dialog displays, through which you select the picture to link to the document. The picture displays at the current cursor location. Linked picture data is not saved with the document, only the filename is stored within the document.	
Embed an Ole object in the text	Position the cursor at the point at which to embed the object, and select the Insert Ole Object menu option. The Insert Object dialog displays, listing the applications that are available to create the object. When you select an application, the editor launches it and you create the required object using this application. When you save the application, the editor inserts an icon that indicates the inserted object. You can later edit the object using the application, by double-clicking on the object. You can also use the Edit Paste Special menu option to import the OLE objects, provided that the object is available in the clipboard.	Block editing 
Edit an embedded picture	Click on the picture and select the Edit Edit Picture menu option. The Edit Current Picture Parameters dialog displays, through which you can change the width and height of the picture, in inches. You can also align the top, bottom, or middle of the picture with the base line of the text.	
Edit an embedded Ole object	Double-click on the icon that indicates the inserted Ole object. Alternatively, position the cursor on the icon and select the Edit Edit Ole Object menu option. The editor opens the object in the application used to create it, and you can edit the object.	

To	Action	See also
Insert a background picture for the text	Select the Other Background Picture menu option. A browser dialog displays, through which you select the bitmap or metafile file to insert as a background picture. The picture occupies the entire text area. To remove the background picture, deselect the Background Picture menu option.	
Insert an RTF File	Position the cursor at the point at which to insert the file, and select the Insert Insert RTF File menu option.	

17.1.1.5.7.6 Character Formatting

When you change the format of existing text, any new characters you type immediately following automatically assume the formatting characteristics of the existing text.

To	Action	See also
Apply character format	Highlight the text to which to apply the format, and use one or more of the following menu options or key combinations, as required: <ul style="list-style-type: none"> • Font Normal, or press (Alt+0) • Font Bold, or press (Ctrl+B) • Font Underline, or press (Ctrl+U) • Font Double Underline, or press (Ctrl+D) • Font Italic, or press (Ctrl+I) • Font Superscript, or press (Alt+4) • Font Subscript, or press (Alt+5) • Font Strike, or press (Alt+6) (puts a line through the text) • Font All Caps • Font Small Caps <p>To reset any character format, highlight the text and select the Font Normal menu option, or press (Alt+0) .</p>	
Change font typeface and point size	Highlight the text to change and select the Font Fonts menu option, or press (Alt+F10) . The Font Selection dialog displays, from which you select the required typeface and point size. Click on the OK button.	
Change character style	Highlight the text to change and select the Font Style menu option. The Select a Style dialog displays, listing the currently-defined character styles in the template stylesheet. Select the required style and click on the OK button.	Template stylesheet ^[1759]
Change the color of text, background (permanent highlight) or underline	Highlight the text to change and select one or more of the following options, as required: <ul style="list-style-type: none"> • Font Text Color • Font Background Color • Font Underline Color <p>In each case, the Color dialog displays, through which you can select or define the required color. When you have selected a color, click on the OK button.</p>	
Change character	Normal character spacing is 20 Twips. If you want to change	

To	Action	See also
spacing	<p>this (or return to it), highlight the text to adjust, and select the Font Spacing menu option. The Character Spacing dialog displays.</p> <p>Select the radio button to expand or compress spacing, or to return to normal spacing. If you are changing from normal spacing, enter the number of Twips to set the spacing to. Click on the OK button.</p>	
Hide text	<p>Hidden text is not displayed on the screen or printer, but remains in the document and is not deleted.</p> <p>Highlight the text to hide and select the Font Hidden menu option, or press (Ctrl+H). The highlighted text is not displayed and the rest of the text closes up.</p> <p>To view hidden text, select the View Hidden Text menu option. You can then make the text normal again by highlighting it and deselecting the Font Hidden menu option.</p>	
Box text	Highlight the text to box and select the Font Boxed menu option. This creates a broken-line border around the selected text.	
Insert a non-breaking space	Move the cursor to the point at which to insert the non-breaking space and select the Insert Non-breaking Space menu option.	
Insert a non-breaking dash	Move the cursor to the point at which to insert the non-breaking dash and select the Insert Non-breaking Dash menu option.	
Insert an optional hyphen	Move the cursor to the point at which to insert the hyphen and select the Insert Optional Hyphen menu option.	

17.1.1.5.7.7 Paragraph Formatting

The functions described below operate on the current paragraph, or on a **highlighted** block of text.

To	Action	See also
Clear all paragraph formatting	Select the Paragraph Normal menu option. The editor pushes the paragraph back up to the page margin.	
Set text flow in document	<p>To set the text flow for:</p> <ul style="list-style-type: none"> • A selected block of text, select the Paragraph Text Flow menu option; the Paragraph Text Flow dialog displays • The entire document, select the Edit Document Text Flow menu option; the Document Text Flow dialog displays. <p>In either case, select the required text flow direction and click on the OK button.</p>	
Center text	Select the Paragraph Center menu option or press (Alt+8) .	
Right-justify text	Select the Paragraph Right Justify menu option or press (Alt	

To	Action	See also
	+9) .	
Justify both sides of text	Select the Paragraph Justify Both menu option.	
Set double line spacing	Select the Paragraph Double Space menu option. A double-spaced paragraph has a blank line between each text line.	
Indent paragraph left	Select the Paragraph Indent Left menu option or press (Alt+L). Select the option again to increase the indent.	
Indent paragraph right	Select the Paragraph Indent Right menu option or press (Alt+R). Select the option again to increase the indent.	
Create hanging indent	Select the Paragraph Hanging Indent menu option or press (Alt+T). Select the option again to increase the indent of all lines below the first.	
Keep paragraph lines together	Select the Paragraph Keep Together menu option. The editor attempts to keep all lines within the paragraph on the same page.	
Keep paragraph with next	Select the Paragraph Keep with Next menu option. The editor attempts to keep the last line of the current paragraph and the first line of the next paragraph on the same page.	
Prevent 'widow' and 'orphan' lines	Select the Paragraph Widow/Orphan Control menu option. The editor attempts to avoid having: <ul style="list-style-type: none"> the first line of the paragraph on the previous page ('widow' line) the last line of the paragraph on the next page ('orphan' line). 	
Start text on new page	Move the cursor to the point at which to start the new page, and select the Paragraph Page Break Before menu option.	
Insert border and shading for text block	Highlight the required text and select the Paragraph Border and Shading menu option. The Paragraph Box Parameters dialog displays, on which you specify which sides of the box to display (including a line between text lines), whether the lines are thick or doubled, the degree of gray shading behind the text, and the color of the lines.	
Define line spacing	Highlight the required lines and select the Paragraph Paragraph Spacing menu option. The Paragraph Spacing Parameters dialog displays, on which you specify the line spacing and the point spacing before and after lines.	
Set a background color for text space	Highlight a text string or block of text and select the Paragraph Background Color menu option. The Color dialog displays, on which you select the background color. The editor highlights the full width of the page in that color, for the selected lines.	
Create a bulleted list	Highlight the required lines of text and select the Paragraph Bullet menu option. The editor formats the lines into a simple bullet list.	
Create a numbered list	Highlight the required lines of text and select the Paragraph Numbering menu option. The editor formats the lines into a	

To	Action	See also
	simple numbered list.	
Apply numbering to paragraphs	Set up a numbering list and overrides (Edit List and Overrides) and apply the numbering levels to the template sections (Paragraph List Numbering).	Apply the numbering levels ^[1762] Numbering list and overrides ^[1761]
Apply a paragraph style from the template stylesheet	Select the Paragraph Style menu option. The Select a Style dialog displays, listing the currently-defined paragraph styles in the template stylesheet. Select the required style and click on the OK button.	Template stylesheet ^[1759]

Learn More:

- [Block Editing](#) ^[1746]

17.1.1.5.7.8 Tab Support

The RTF Style Template Editor supports *left*, *right*, *center* and *decimal* tabs. Tabs are very useful for creating columns and tables. A paragraph can have as many as twenty tab positions.

A tab usually applies to every line of the current paragraph. However, if you highlight a block of text before setting a tab, the tab then applies to every line in the highlighted text.

Topics:

Topic	Detail	See also
	<p>You can create tabs quickly and easily using the ruler at the top of the screen. To create a:</p> <ul style="list-style-type: none"> • Left tab, click on the required tab position on the ruler; the left tab is indicated on the ruler by an L shape. • Right tab, right-click on the required tab position on the ruler; the right tab is indicated on the ruler by a reversed L shape • Center tab, press (Shift) and click on the required tab position on the ruler; the center tab is indicated on the ruler by an inverted T shape • Decimal tab, press (Shift) and right-click on the required tab position on the ruler; the decimal tab stop is indicated on the ruler by an inverted T shape with a dot on the right hand side. This tab is for numbers with a decimal point; numbers scroll left from the tab until you type a point, then numbers scroll right <p>You can also set tabs using the Paragraph Set Tab menu option, which displays the Set a Tab Position dialog. This enables you to specify the tab type, and provides two advantages over the ruler: you can set the tab position with more precision and with a clear value that you can duplicate; and you can add a tab leader line (dot, hyphen, or underline).</p>	

Topic	Detail	See also
	<ul style="list-style-type: none"> To clear a single tab position for selected text, select the Paragraph Clear Tab menu option. The Clear a Tab Position dialog displays, on which you select the tab to clear To clear all tab stops for selected text, select the Paragraph Clear All Tabs menu option To move a tab position using the mouse, click on the tab symbol on the ruler and drag it to the new position 	

Notes:

- The **Other | Snap To Grid** menu option affects the movement of the tabs (and the paragraph indentation markers) on the ruler. When you select this option, the movements of the tab markers are locked on to an invisible grid at intervals of 1/16 inch (half a ruler division)

17.1.1.5.7.9 Page Breaks and Repagination

You can force a page break in the document by selecting the **Insert | Insert Break | Page Break** menu option, or by pressing (**Ctrl+Enter**). The forced page break is indicated by a solid line in the editing window.

If **Page Mode** is off, the editor also marks automatic page breaks when the text overflows a page; these are indicated by a dotted line.

You can repaginate your document, using the **Edit | Repaginate** menu option. This updates the **Page Number** and **Page Count** fields, and recompiles the table of contents.

You insert the **Page Number** and **Page Count** fields as follows:

Topic	Detail	See also
Insert the page number	Position the cursor at the point at which to display the page number, and select the Insert Page Number menu option. The page number is displayed in gray.	
Insert the page count	Position the cursor at the point at which to display the total number of pages in the document, and select the Insert Page Count menu option. The page count is displayed in gray.	

17.1.1.5.7.10 Headers and Footers

To	Action	See also
Edit the page header and footer text	<p>In Page Mode, select the Edit Edit Page Header/Footer menu option. A paragraph marker displays at the top and bottom of each page, and you can type in, format or delete the appropriate text.</p> <p>If Page Mode is turned off, all page headers and footers are displayed in a block at the start of the document, with identifying labels. You can also edit the text here.</p> <p>Each section in a document can have its own page header and footer.</p>	
Create the header and footer for the initial page of the document	In Page Mode, select the Edit Edit Page Header/Footer menu option and then select the Edit First Page Header/Footer Create First Page Header or Create First Page Footer menu option. A paragraph marker displays at the top or bottom of the first page, and you can type in and format the appropriate text.	

To	Action	See also
	If Page Mode is turned off, all page headers and footers are listed in a block at the start of the document, with identifying labels. You can also edit the text here.	
Delete the header and footer for the initial page of the document	<p>Whilst you can delete the first page header or footer text by simple editing, you must specifically delete the 'first page' assignment in order to display the header and footer text of the next section on the first page.</p> <p>In Page Mode, select the Edit Edit Page Header/Footer menu option and then select the Edit First Page Header/Footer Delete First Page Header or Delete First Page Footer menu option. This removes the first page text and assignment, and displays the next-defined header and footer text.</p>	
Create a footnote	<p>Move the cursor to the position at which to insert the footnote marker, and select the Insert Footnote/Endnote Footnote menu option. The Footnote Parameters dialog displays, on which you enter the footnote marker and footnote text, and select whether to make the marker a superscript.</p> <p>Click on the OK button. The editor inserts the footnote marker at the current cursor location and, in Page Mode, displays the footnote text at the bottom of the page.</p>	
Edit footnote text	<p>Select the Edit Edit Footnote/Endnote Edit Footnote Text menu option. The text of each footnote displays in the document text where its marker was inserted. Locate the text and make the required changes. In Page Mode, the modified footnote displays at the bottom of the page.</p> <p>When you have finished editing footnote text, <i>deselect</i> the Edit Edit Footnote/Endnote Edit Footnote Text menu option. The footnote text is no longer shown in the document text.</p>	
Create an endnote	<p>Move the cursor to the position at which to insert the endnote marker, and select the Insert Footnote/Endnote Endnote menu option. The Endnote Parameters dialog displays, on which you enter the endnote marker and endnote text, and select whether to make the marker a superscript.</p> <p>Click on the OK button. The editor inserts the endnote marker at the current cursor location and, in Page Mode, displays the endnote text at the end of the section or document.</p>	
Edit endnote text	<p>Select the Edit Edit Footnote/Endnote Edit Endnote Text menu option. The text of each endnote displays in the document text where its marker was inserted. Locate the text and make the required changes. In Page Mode, the modified endnote displays at the bottom of the page.</p> <p>When you have finished editing endnote text, <i>deselect</i> the Edit Edit Footnote/Endnote Edit Endnote Text menu option. The endnote text is no longer shown in the document text.</p>	

17.1.1.5.7.11 Hyperlinks and Bookmarks

Topic	Detail	See also
Manage bookmarks	Each template contains a number of bookmarks that mark the sections. You can apply these bookmarks to related sections, create and assign your own, delete those that are not required, and locate	

Topic	Detail	See also
	<p>specific bookmarks in the document.</p> <p>In Page Mode, select the Insert Bookmark menu option. The Bookmark dialog displays.</p> <ul style="list-style-type: none"> • To assign a bookmark to the current cursor position, either type a new bookmark in the top field or select one from the list, and click on the Insert button. • To delete an existing bookmark, click on it in the list and click on the Delete button. • To mark a bookmark with the number of the page it is on, click on the bookmark in the list and click on the Set Page Reference button. • To locate a bookmark in the text, click on it in the list and click on the Go to button. 	
Insert hyperlinks	<p>Right-click on the point at which to create the hyperlink and select the Insert Hyperlink context menu option. The Insert Hyperlink dialog displays. You can create a hyperlink within an RTF document to an external document, Help topic or web page.</p> <p>In the Link Text field type the text to be hyperlinked, and in the Link Code field type or paste the web page URL, help topic file or external file path and name.</p> <p>To capture the help topic file name, right-click on the displayed topic, select the Properties context menu option, and copy the file name. When you insert the file name in the Link Code field, ensure that the file name has the prefix <code>\$Help://</code>.</p> <p>Click on the OK button. The hyperlinked text displays in the document. Double-click on the link to display the web page or external document.</p>	

17.1.1.5.7.12 Table Commands

The **Table** menu enables you to create a new table, or to edit an existing table's attributes.

To	Action	See also
Insert a table in the document	<p>Position the cursor at the appropriate point, and select the Table Insert Table menu option. The New Table Parameters dialog displays, in which you specify the number of table rows and columns.</p> <p>The editor initially creates cells of equal width. You can, however, change the cell width by dragging the cell borders using the mouse. When Page Mode is deselected, the table structure is not visible.</p>	
Add a header row	<p>Select the rows to act on, and then select the Table Header Row menu option. Apply any heading settings and formatting to the rows in the highlighted block.</p>	
Insert a new row above the current row	<p>Select the Table Insert Row menu option.</p>	
Insert a new column to the left of the current column	<p>Select the Table Insert Column menu option.</p>	

To	Action	See also
Merge cells	Select the cells to merge and select the Table Merge Cells menu option. The width of the resulting cell is equal to the sum of the merged cells. You can merge cells across a row, down a column, and in a block spanning both rows and columns.	
Split a cell	Select the cell to split and select the Table Split Cell menu option. The selected cell is split into two cells of equal width. Any text in the original cell is assigned to the first cell. The second cell is created empty.	
Delete cells	Select the cells to delete and select the Table Delete Cells menu option. The Delete Table Cells dialog displays, on which you specify whether to delete: <ul style="list-style-type: none"> • Cells - deletes the highlighted cells • Columns - deletes all the cells in the highlighted column or columns • Rows - deletes all the cells in the highlighted row or rows. If you delete all cells in a table, the table itself is automatically deleted.	
Position the table on the page	Click on any part of the table and select the Table Row Position menu option. The Table Row Alignment dialog displays, on which you select to left-align, center or right-align the table on the page. This option has little effect if the table is wide enough to span the page or text column.	
Set the height of a row, or all rows	Select the row to adjust and select the Table Row Height menu option The Row Height Parameters dialog displays, enabling you to set an automatic row height, a minimum row height, or an exact row height. You can apply the setting to the selected rows only, or to all rows in the table.	
Keep row text together if it continues over a page	Select the rows to protect (preferably all rows in the table) and select the Table Keep Row Together menu option. If the row continues over the end of the page, the whole row is moved to the top of the next page.	
Set text flow in rows	Select the rows and select the Table Row Text Flow menu option. The Table Text Flow dialog displays, on which you select the direction of flow of the text and select to apply the setting to the selected rows or all rows in the table. This option also moves the whole row over to the appropriate side of the page or column.	
Set the width of selected cells	Select the cells to act on and select the Table Cell Width menu option. The Set Cell Width dialog displays, on which you set the cell width and text margin and apply them to: <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Define the cell border width	Select the cells to act on and select the Table Cell Border Width menu option. The Set Cell Border dialog displays, on which you set the width of the line at any or all of the top, bottom, left and right of a cell, or whether to draw a uniform border around the cells. You can also set the text margin, and apply all the settings to: <ul style="list-style-type: none"> • All cells in a highlighted block 	

To	Action	See also
	<ul style="list-style-type: none"> • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Define the cell border color	<p>Select the cells to act on and select the Table Cell Border Color menu option. The Set Cell Border Color dialog displays, on which you set the color of the line at any or all of the top, bottom, left and right of a cell, or whether to have a uniformly colored border around the cells. You then apply the settings to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Define the cell shading	<p>Select the cells to act on and select the Table Cell Shading menu option. The Cell Shading Parameters dialog displays, on which you set the shading percentage. The value 0 indicates the palest background, whereas the value 100 indicates a black background. You then apply the setting to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Define the cell background color	<p>Select the cells to act on and select the Table Cell Color menu option. The Cell Color Parameters dialog displays, on which you set the cell background color. You then apply the color to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Vertically align cells	<p>Select the cells to act on and select the Table Cell Vertical Align menu option. The Cell Vertical Alignment dialog displays, on which you select to align the selected cells by top, center, bottom or baseline. You then select to align:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Rotate cell text	<p>Select the cells to act on and select the Table Cell Rotate Text menu option. The Cell Text Rotation dialog displays. On which you select to display text horizontally across the cell, vertically up the cell, or vertically down the cell. You then select to apply the rotation to:</p> <ul style="list-style-type: none"> • All cells in a highlighted block • The selected cells only • All cells in the selected column or columns, or • All cells in the selected row or rows. 	
Select column	<p>Click on a cell and select the Table Select Current Column menu option. The whole column is highlighted and selected for further formatting.</p>	
Show / hide table outline	<p>Click on a table cell and select the Table Show Gridlines menu option. This displays or hides the grid lines around the table cells. The</p>	

To	Action	See also
	grid lines are for display purpose only and do not appear on the printed document.	

17.1.1.5.7.13 Sections and Columns

The editor enables you to divide a document into multiple sections. A multiple section document is useful to:

- Vary the page margins from one page to another
- Create multiple columns of text.

To	Action	See also
Create a new section	Select the Insert Insert Break Section Break menu option. This creates a new section on a new page. This option is not available when Edit Edit Page Header/ Footer is active.	
Edit the section parameters	Select the Edit Edit Section menu option. The Section Parameters dialog displays. Define: <ul style="list-style-type: none"> • The number of columns and column spacing; text in a multiple column section wraps at the end of the column. When the text reaches the end of the page, it resumes in the next column. When Page Mode is off, the page contains a single very thin and long column. When Page Mode is on, the correct column layout is shown. • The orientation - Portrait or Landscape • Whether to start the new section on the next page • The direction of text flow • Any special printing characteristics for the section. You can also define any special page margins by selecting the File Page Layout menu option.	File and Print Options ^[1745]
Delete a section break	Move the cursor onto the section break line and press (Delete).	
Create a column break	Move the cursor to the appropriate point in the text and select the Insert Insert Break Column Break menu option. Normally in a multiple column section, the text flows from the end of one column to the top of the next column. A column break forces the text to the next column before the current column is completely filled. A column break is indicated by a line with a 'dot and dash' pattern. To delete the column break, simply position the cursor on the column break line and press (Delete).	

17.1.1.5.7.14 Stylesheets and Table of Contents

The editor supports *Character* and *Paragraph-type* stylesheet style items:

- The Character stylesheet style constitutes a set of character formatting attributes and is applied to a character string
- The Paragraph stylesheet style constitutes both a set of character formatting attributes and a set of paragraph formatting attributes, and is applied to one or more paragraphs

You can also include special, structured text in the document, such as **page number**, **date and time** and **text input** fields.

To	Action	See also
Use double-byte characters	Select the Edit Inline Ime menu option This enables you to enter double-byte characters without using an external IME application	
Create and edit styles	Select the Edit Edit Style menu option The Edit Stylesheet dialog displays Select the appropriate radio button to define a character style or a paragraph style Either select an existing style to modify from the list box, or type in a name for a new style; click on the OK button to begin recording the style properties You can use the ruler, toolbar or menu selections to modify the style items; these also reflect the currently-selected properties for the stylesheet item - please note that the paragraph properties are enabled only for the paragraph stylesheet items After you have defined the required style, either select the Edit Edit Style menu option again or click anywhere in the document: <ul style="list-style-type: none"> • If you modified an existing stylesheet item, the document automatically reflects the updated style • If you created a new stylesheet item, you can apply the style to highlighted text by selecting the Font Style or Paragraph Style menu options 	
Apply character styles	Select the Font Style menu option to apply a stylesheet style to a highlighted character string	
Apply paragraph styles	Select the Paragraph Style menu option to apply a stylesheet style to a highlighted paragraph or range of paragraphs	
Insert a table of contents	Create and apply the required heading styles using the Edit Edit Style menu option, as above Move the cursor to the point at which to insert the table of contents and select the Insert Table of Contents menu option The table of contents is automatically updated whenever repagination occurs	Repagination n ₁₇₅₄
Insert date and time fields	Move the cursor to the point at which to insert the current date and time, and select the Insert Date and Time menu option The Insert Current Date/Time dialog displays, from which you select the required date and time format The date and time are automatically updated whenever the page text is refreshed	
Insert your own data fields	Not Supported In The Enterprise Architect RTF Generator Move the cursor to the point at which to insert the data field and select the Insert Data Field menu option The Data Field Parameters dialog displays, in which you enter the field name and data value	

To	Action	See also
Insert your own text entry field	<p>Move the cursor to the point at which to insert the text entry field and select the Insert Text Input Field menu option</p> <p>The Input Field Parameters dialog displays, in which you enter:</p> <ul style="list-style-type: none"> • The field name • The initial value to display as a default • The maximum length of the field • The text font in which to display the text <p>You also specify whether or not the field has a border</p>	
Insert a selectable checkbox	<p>Move the cursor to the point at which to insert the checkbox and select the Insert Checkbox Field menu option</p> <p>The Checkbox Field Parameters dialog displays, in which you enter:</p> <ul style="list-style-type: none"> • The field name • Whether it defaults to selected • The size of the box surrounding the check 	
Define level numbering in generated document	<p>Select the Edit List and Overrides menu option</p> <p>Set up the numbering list and the list overrides</p> <p>Apply the numbering list to the headings set for packages and elements, using paragraph numbering</p>	User defined list numbering [1761]

17.1.1.5.7.15 User-Defined Section Numbering

You might want to define the numbering format for the section levels in your generated RTF document.

For example:

- ```

1. Package Level 1
 1.1 Package Level 2 (child package)
 1.1.1 Element Level 1
 1.1.1.1 Element (child element)

```

To define the numbering format you first create a numbering list and then create a set of list overrides for this list. The overrides must also have the initial 1.0.0 setting altered to 1.1.1. You can then apply the numbering list to the headings set for packages and elements, using paragraph numbering.

(See **Section Numbering in Virtual Documents** for further information on applying continuous section numbering throughout a **Virtual Document**.)

To define the numbering format, follow the steps below

#### How To:

| Step | Action                                                                                                                                                  | See Also |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | In the <b>Template Editor</b> , select the <b>Edit   List and Overrides   Create List Item</b> menu option. The <b>List Properties</b> dialog displays. |          |
| 2.   | In the <b>List Name</b> field, type a name for the list. Click on the <b>OK</b> button to close the dialog.                                             |          |
| 3.   | Select the <b>Edit   List and Overrides   Create List Override</b> menu option. The <b>List Override Properties</b> dialog displays.                    |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                              | See Also |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 4.   | In the <b>List to Override</b> field, type or select the name of the list you have just created. Click on the <b>OK</b> button to close the dialog.                                                                                                                                                                                                                 |          |
| 5.   | To set the list level properties for each level, select the <b>Edit   List and Overrides   Edit List Level</b> menu option. The List Level Properties dialog displays.                                                                                                                                                                                              |          |
| 6.   | To set the first level numbering (used in the Package Section), select the <b>List override</b> radio button and type or select the <b>list override</b> item you have just created.                                                                                                                                                                                |          |
| 7.   | Ensure that the <b>List Level</b> field is set to <b>1</b> (for Packages) and the <b>Number text</b> field is set to ~1~. Click on the <b>OK</b> button to save the values and close the dialog.                                                                                                                                                                    |          |
| 8.   | Open the dialog again ( <b>Edit   List and Overrides   Edit List Level</b> ) and set: <ul style="list-style-type: none"> <li>• <b>List Level</b> to <b>2</b> (for the <i>Element Section</i> or <i>Child Package Section</i>, for example)</li> <li>• <b>Start at</b> to <b>1</b> (to ensure that numbering at this level begins at 1.1 rather than 1.0)</li> </ul> |          |
| 9.   | Click on the <b>OK</b> button to close the dialog and save the changes.                                                                                                                                                                                                                                                                                             |          |
| 10.  | Repeat steps 5 to 9 as required, incrementing the list level number and resetting <b>Start</b> at to <b>1</b> each time.                                                                                                                                                                                                                                            |          |

To apply the numbering levels you have defined (above), follow the steps below

#### How To:

| Step | Action                                                                                                                                                                                                                                                                                                                                                    | See Also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | In the <b>Content</b> window of the <b>Template Editor</b> , select the first item of text to be numbered (for example, <i>Package</i> ).<br><br><pre> package-&gt;  Package: {Pkg.Name}   element-&gt;  Element: {Element.Name}   child-elements-&gt;  &lt;-child-elements  &lt;-element  child-packages-&gt;  &lt;-child-packages  &lt;-package  </pre> |          |
| 2.   | Set the text style, using the style drop-down field in the <b>Template Editor</b> toolbar.<br><br><pre> package-&gt;  Package: {Pkg.Name} ¶ </pre>                                                                                                                                                                                                        |          |
| 3.   | Click on the <b>Paragraph   List Numbering</b> menu option. The <b>Apply paragraph numbering using Lists</b> dialog displays                                                                                                                                                                                                                              |          |
| 4.   | Select the required Numbering List and Override, and set the <b>Level</b> field                                                                                                                                                                                                                                                                           |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <p>to the required level (1, for the top level). Click on the <b>OK</b> button to close the dialog, and check that the required level has been applied to the selected text.</p> <p><b>package&gt;</b></p> <p><b>1. Package:-{Pkg.Name}</b></p>                                                                                                                                                                                                                                                                                                                                                          |          |
| 5.   | <p>Repeat steps 1 to 4 for the next level (Element), but at step 4 change the <b>Level</b> field to 2.</p> <p><b>element&gt;</b></p> <p><b>1.1 Element:-{Element.Name}</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                           |          |
| 6.   | <p>Continue applying the overrides for lower section levels as necessary, then save the template and generate your RTF report. The output should now resemble the following example:</p> <p><b>1. Package: Formal Requirements</b></p> <p><b>1.1 Package: Manage Users</b></p> <p><b>1.1.1 Element: REQ011 - Manage User Accounts</b></p> <p><b>1.1.2 Element: REQ016 - Add Users</b></p> <p><b>1.2 Package: Manage Inventory</b></p> <p><b>1.2.1 Element: REQ019 - Manage Inventory</b></p> <p><b>1.2.2 Element: REQ020 - Receive Books</b></p> <p><b>1.2.3 Element: REQ021 - List Stock Levels</b></p> |          |

**Learn More:**

- [Section Numbering in Virtual Documents](#)<sup>[1782]</sup>
- [Virtual Document](#)<sup>[1774]</sup>

**17.1.1.5.7.16 Frames and Drawing Objects**

A frame is a rectangular area that can contain both text and pictures on the page. The text *outside* the frame flows around it. A drawing object can be a text box, rectangle or a line. The drawing object *overlays* the text around it. You can see frames and drawing objects only if **View | Page Mode** is selected. However, the *content* of a frame or text box is still visible if **Page Mode** is deselected.

| To                                                      | Action                                                                                                      | See also |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|----------|
| Embed a frame or drawing object at the cursor position  | Select the <b>Insert   Frame</b> or <b>Insert   Drawing Object</b> menu option.                             |          |
| Insert text into the frame or drawing object text box   | Click inside the outline and type the text at the cursor position.                                          |          |
| Rotate text to display it down the side of the frame or | Select the <b>Edit   Edit Frame/Drawing Object   Rotate Text</b> menu option and select the text direction. |          |

| To                                                                                                           | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See also |
|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| text box                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |
| Insert a picture into a frame or drawing object text box                                                     | Copy the picture, click inside the outline and paste the picture at the cursor position.                                                                                                                                                                                                                                                                                                                                                                                                      |          |
| Size a frame or drawing object                                                                               | Click inside the outline, click on a sizing tab on the outline and drag the tab to the required position.<br><br>If the frame or text box contains only a picture, the picture size is automatically adjusted to fill the outline. Any text inside the outline is automatically wrapped to adjust to the new width.<br><br>In a frame the frame height is automatically adjusted, if necessary, to enclose all lines. In a text box, you must adjust the height manually to enclose the text. |          |
| Move the frame or drawing object                                                                             | Click inside the outline and then move the cursor just outside the outline so that the cursor changes to a plus-shape. Drag the plus shape (and hence the outline) to the new location.                                                                                                                                                                                                                                                                                                       |          |
| Edit the relationship between a frame or drawing object and a point on the page (the vertical base position) | Click on the outline and select the <b>Edit   Edit Frame/Drawing Object   Vertical Base Position</b> menu option. Select the point to lock the outline to. Outlines locked to the top of the page or the top of the margin retain their vertical position when you insert text before them.                                                                                                                                                                                                   |          |
| Edit the border and the background of a drawing object                                                       | Select the <b>Edit   Edit Frame/Drawing Object   Edit Drawing Object</b> menu option. On the <b>Line and fill attributes</b> dialog, select the options for the preferred border, line color, fill color and wrapping effect on the template text.                                                                                                                                                                                                                                            |          |
| Delete a frame or drawing object                                                                             | Click on the outline and press ( <b>Delete</b> ). The editor prompts you to confirm the deletion. Click on the <b>OK</b> button. Note that the deletion is actually reversible - press ( <b>Ctrl+Z</b> ) or select the <b>Edit   Undo</b> menu option.                                                                                                                                                                                                                                        |          |

#### 17.1.1.5.7.17 Search and Replace Commands

The first three menu options below all invoke the Search String Parameters dialog, if no search term has been defined.

Specify the term to search for, whether to search from the start of the file or forwards or backwards from the current cursor position, and whether the search should exactly match the case of the search term.

| To                                                         | Action                                                                                                                                                                                                                                    | See also |
|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Search for a text string</b>                            | Select the <b>Other   Search</b> menu option, or press ( <b>F5</b> )<br><br>The Search String Parameters dialog displays<br><br>The editor searches for the first instance of the specified character string as defined by the parameters |          |
| <b>Find the next instance of a previously-defined text</b> | Select the <b>Other   Search Forward</b> menu option, or press ( <b>Ctrl+F</b> )<br><br>The editor searches forwards for the next instance of the specified text string in the file, and highlights it                                    |          |

| To                                                                                  | Action                                                                                                                                                                                                                                                                                                                                                                                                                               | See also |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| string in the file                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                      |          |
| <b>Find the previous instance of the previously-defined text string in the file</b> | Select the <b>Other   Search Backward</b> menu option, or press <b>( Ctrl+Shift+F )</b><br><br>The editor searches backwards for the previous instance of the text string in the file, and highlights it                                                                                                                                                                                                                             |          |
| <b>Replace a text string</b>                                                        | Select the <b>Other   Replace</b> menu option, or press <b>( F6 )</b><br><br>The Replace String Parameters dialog displays, in which you specify: <ul style="list-style-type: none"> <li>• The text string to locate</li> <li>• The text string to replace it with</li> <li>• Whether to search the whole document or a highlighted block of text</li> <li>• Whether to confirm each replacement before making the change</li> </ul> |          |

### 17.1.1.6 Import RTF Template

Enterprise Architect provides a number of RTF document templates, and enables you to create others. However, you might already have corporate formats and templates in use in your organization, so Enterprise Architect also enables you to import existing templates into the RTF Generator.

To import an external template, follow the steps below

#### How To:

| Step | Action                                                                                                                                                                                                                   | See Also                                                                                                                                |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | Save the external template as an RTF document                                                                                                                                                                            |                                                                                                                                         |
| 2.   | In Enterprise Architect, <b>create a new blank template</b> . Name the template but do not specify an existing template to copy from.                                                                                    | <a href="#">RTF Templates Tab</a>                  |
| 3.   | When the template is listed on the <b>Templates</b> tab of the <b>Generate RTF Documentation</b> dialog, click on the name and click on the <b>Edit</b> button. The <b>RTF Document Template Editor</b> dialog displays. |                                                                                                                                         |
| 4.   | Select the <b>File   Import</b> menu option. The Microsoft Word file <b>Open</b> dialog displays.                                                                                                                        |                                                                                                                                         |
| 5.   | Locate your template RTF file, and click on the <b>Open</b> button. The <b>Open</b> dialog closes, returning you to the <b>Document Template Editor</b> dialog. This now contains your imported template.                |                                                                                                                                         |
| 6.   | Select the <b>File   Save</b> menu option. If necessary, make any changes to the template and select <b>File   Save</b> again before selecting <b>File   Close</b> to exit the dialog.                                   | <a href="#">RTF Style Template Editor Options</a>  |

#### Notes:

- Standard graphical images (such as a logo in the header, main text or footer) are imported. However, any Word-based meta-file graphics are not imported
- You can select the new template to use in generating an RTF document, either on the **Generate RTF Documentation** dialog or in a **Master Document** or **Model Document** element

**Learn More:**

- [Generate RTF Documentation](#)<sup>[1727]</sup>
- [Master Document](#)<sup>[1776]</sup>
- [Model Document](#)<sup>[1777]</sup>

**17.1.1.7 Resource Documents**

The *Resource Document* feature enables a particular documentation configuration to be 'remembered', linking the loaded template within the **Generate RTF Documentation** dialog to the current highlighted package. If a particular template is always used with a specific package, and multiple cases of documentation exist to be propagated, saving these as Resource Documents can ease document generation later.

To create and use Resource Documents, follow the steps below

**How To:**

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See Also                                                 |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| 1.   | Open the <b>Generate RTF Document</b> dialog.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <a href="#">Generate RTF Documents</a> <sup>[1727]</sup> |
| 2.   | Click on the <b>Resource Document</b> button. The <b>Save current as document definition</b> dialog displays                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                          |
| 3.   | In the <b>Enter Value</b> field, type a name for the document and click on the <b>OK</b> button. The document is added to the <b>Resources</b> window for easy future access (as for the <i>usecasedoc</i> entry in the illustration below).                                                                                                                                                                                                                                                                                                                                          |                                                          |
| 4.   | To generate documentation from the <b>Resources</b> window, right-click on the required document. The context menu displays.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                          |
| 5.   | Select the required option.<br>The context menu options are: <ul style="list-style-type: none"> <li>• <b>Open Document</b> - Opens the corresponding .RTF file, as specified by the RTF template Filename property</li> <li>• <b>Generate Document</b> - Opens the <b>Generate RTF Documentation</b> dialog, loaded with the specified template</li> <li>• <b>Auto Generate Document</b> - Generates documentation, with the document located at the path specified by the template's Filename property</li> <li>• <b>Delete Document</b> - Removes the specified document</li> </ul> |                                                          |

| Topic                                    | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Batch Generate Resource Documents</b> | To generate a number of RTF documents at the same time, right-click on the <i>RTF Documents</i> folder name and select the <b>Generate Documents</b> context menu option. The <b>Batch Document Generation</b> dialog displays.<br><br>The dialog lists all resource documents in the <i>RTF Documents</i> folder, defaulted to selected. Deselect the checkbox against each document that you do not want to generate (or click on the <b>Deselect All</b> button to clear all selections, then select those you require). Click on the <b>OK</b> |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                       | See also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>button to generate each of the remaining reports into their respective target file locations.</p> <p>The <b>Generate All Documents</b> context menu option automatically generates every document in the <i>RTF Documents</i> folder, without displaying the <b>Batch Document Generation</b> dialog.</p> |          |

**Notes:**

- Resource Documents, saved into the **Resources** window, save the package, output file destination and template name from the **Generate** tab of the Generate RTF Documentation dialog, plus the data separately defined in the Options, Advanced and Element Filter tabs of the dialog

**Learn More:**

- [Generate RTF Documentation Dialog](#) <sup>[1732]</sup>
- [Document Options](#) <sup>[1767]</sup>
- [Element Filter](#) <sup>[1770]</sup>

### 17.1.1.8 Document Options

The RTF report options enable you to set type filters and order the elements.

**Topics:**

| Topic                          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                                                                                                                                                                                                                                                                           |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>How to Access</b>           | <p>You can access the options from two different places; the start point affects the persistence of options selected:</p> <ul style="list-style-type: none"> <li>• If you access the options on the Options and Advanced tabs of the Generate RTF Documentation dialog, you can define settings for the current report to be run; selections are non-persistent, and are reset when you exit the dialog or select a different template</li> <li>• If you access the options on the Document Options dialog, by clicking on the <b>File   Document Options</b> menu option on the RTF Style Template Editor dialog, the settings are saved with the template as the default settings for any run of this report</li> </ul> | <p><a href="#">Generate RTF Documentation</a> <sup>[1732]</sup></p> <p><a href="#">File and Print Options</a> <sup>[1745]</sup></p> <p><a href="#">RTF Style Template Editor</a> <sup>[1738]</sup></p> <p><a href="#">RTF Element Filters</a> <sup>[1770]</sup></p> <p><a href="#">Other</a> <sup>[1771]</sup></p> |
| <b>Document Options dialog</b> | <p>The Document Options dialog provides the options from both of the Options and Advanced tabs of the Generate RTF Documentation dialog, plus the Element Filters and Other Filters tabs, except for:</p> <ul style="list-style-type: none"> <li>• The <b>Switch generator</b> button</li> <li>• The <b>Disable large OLE file support</b> option</li> <li>• The <b>Insert page breaks when generating a Master Document</b> option</li> </ul> <p>The dialog also contains the <b>Optimize for Open Office</b> option, as described for the Generate RTF Documentation dialog, and tabs for setting Element Filters and Other Filters</p>                                                                                 | <p><a href="#">Generate RTF Documentation</a> <sup>[1732]</sup></p> <p><a href="#">Element Filters</a> <sup>[1770]</sup></p> <p><a href="#">Other Filters</a> <sup>[1771]</sup></p>                                                                                                                                |

| Topic | Detail                                             | See also |
|-------|----------------------------------------------------|----------|
|       | Click on the <b>OK</b> button to save your changes |          |

**Reference:**

The Options tab of the Generate RTF Documentation dialog has the following fields:

| Field                         | Usage                                                                                                                                                                                                                                                                                                                                                             | See also |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Only include objects</b>   | Filter elements according to date created or modified. In the first two fields, select the qualifiers from the drop-down lists. In the third field, select the appropriate date.                                                                                                                                                                                  |          |
| <b>Where Package Phase</b>    | Filter elements according to the value of the <b>Package Phase</b> field. In the first field select the qualifier, and in the second type the required phase (or leave the default value <b>All</b> ).                                                                                                                                                            |          |
| <b>With element status</b>    | Filter elements according to status. In the first field, select the qualifier ( <b>like</b> , <b>not like</b> , <b>in</b> , <b>not in</b> ) and in the second field type the value to be used. Values should be enclosed in quotes; for example: "Proposed". If you type more than one value, separate them with a comma; for example: "Proposed", "Implemented". |          |
| <b>Connector Direction</b>    | Filter connectors according to direction.<br><br>If you select <b>Both</b> , the connector is documented twice; once for the source element and once for the target. For the remaining two values, the connector is documented only for the source or target element, as appropriate.                                                                             |          |
| <b>Packages by</b>            | Order packages in the generated documentation in either ascending or descending order of Name, Tree Order, Modified Date or Created Date.                                                                                                                                                                                                                         |          |
| <b>Elements by</b>            | Order elements in the generated documentation in either ascending or descending order of Name, Tree Order, Modified Date or Created Date.                                                                                                                                                                                                                         |          |
| <b>Diagrams by</b>            | Order diagrams in the generated documentation in either ascending or descending order of Name, Tree Order, Modified Date or Created Date                                                                                                                                                                                                                          |          |
| <b>Exclude details for</b>    | Exclude all elements of the selected type or types from the generated document                                                                                                                                                                                                                                                                                    |          |
| <b>Exclude connector type</b> | Exclude all connectors of the selected type or types from the generated document                                                                                                                                                                                                                                                                                  |          |

The **Advanced** tab of the **Generate RTF Documentation** dialog has the following fields:

| Field                            | Usage                                                     | See also |
|----------------------------------|-----------------------------------------------------------|----------|
| <b>Hide 'note-less' elements</b> | Exclude all elements without notes from the documentation |          |

| Field                                                                    | Usage                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also                                                              |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <b>Diagram Format</b>                                                    | Set the diagram format for the images included within the documentation to either Metafile or Bitmap                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                       |
| <b>Skip root package</b>                                                 | Exclude the parent package from the documentation and include only the child packages                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                       |
| <b>No bookmarks</b>                                                      | Stop RTF bookmarks being inserted into the generated document                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                       |
| <b>Adjust Heading Levels</b>                                             | <p>Enable the RTF Generator to automatically restrict the levels of heading generated for nested sub-packages in a document</p> <p>The generator reproduces heading levels down to the value you set; for example, if you have four nested levels of sub-packages and you set this field to:</p> <ul style="list-style-type: none"> <li>• <b>Heading 2</b>, all sub-packages in the report are documented under level 2 headings</li> <li>• <b>Heading 4</b>, the first level of subpackages are documented under level 2 headings, the next level under level 3 headings, and the remainder all under level 4 headings</li> <li>• <b>Heading 6</b>, the first level of subpackages are documented under level 2 headings, the next level under level 3 headings, the next under level 4 headings, and the next under level 5 headings; if you added further levels of sub-package they would all be documented under level 6 headings</li> </ul> <p>The field defaults to <b>Heading 9</b> to accommodate the maximum number of levels of nested subpackages</p> |                                                                       |
| <b>Hide &lt;Anonymous&gt; elements</b>                                   | Hide anonymous elements in the documentation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                       |
| <b>Use style defined in template for notes</b>                           | Override the character formatting specified in your <b>Notes</b> fields with the formatting set for the field in the template                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                       |
| <b>Disable large OLE file support</b>                                    | Disable support for large Object Linking and Embedding (OLE) files                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                       |
| <b>Insert page breaks when generating a Master Document</b>              | Insert a page break after each Model Document in a Master Document                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <a href="#">Model Document in a Master Document</a> <sup>[1774]</sup> |
| <b>Include child elements even if the parent element is filtered out</b> | Document all child elements that are not otherwise filtered out by the restrictions, even if the parent elements are filtered out                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                       |
| <b>Hide 'note-less' connectors</b>                                       | Exclude all connectors without notes from the documentation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                       |
| <b>Switch generator</b>                                                  | <p>Switch from this Generate RTF Documentation dialog (the Enhanced Template Driven Generator) to the Rich Text Format Report dialog (Legacy Generator)</p> <p>This button is not available if you displayed the dialog from the Diagram List, Package Browser or Model Search</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <a href="#">Rich Text Format Report dialog</a> <sup>[1787]</sup>      |

### 17.1.1.9 Element Filters

#### Topics:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                                                                                                                                                                                                                               |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>How to Access</b> | <p>As with the Document Options tabs, you can access the filter details from two different places; the start point affects the persistence of the filter definition.</p> <ul style="list-style-type: none"> <li>If you define the filters on the Element Filters tab of the <b>Generate RTF Documentation</b> dialog, you can create filter settings for the current report to be run. Selections are non-persistent, and are reset when you close the dialog or select a different template.</li> <li>If you access the filter definitions by clicking on the <b>File   Document Options</b> menu option on the <b>RTF Style Template Editor</b> dialog, the settings are saved with the template as the default settings for any run of this report.</li> </ul> <p>You add filters by clicking on the <b>Add Filters</b> button. To edit the filters, either double-click on the panel contents or click on the Edit Filter button, to display the Edit Filters dialog. The format is the element field name, the conditions placed on the field value, any actual value or delimiting value to search on, and whether the filter item is required (mandatory).</p> | <p><a href="#">Generate RTF Documentation</a><sup>[1732]</sup></p> <p><a href="#">File and Print Options</a><sup>[1745]</sup></p> <p><a href="#">RTF Style Template Editor</a><sup>[1736]</sup></p> <p><a href="#">Add Filters</a><sup>[487]</sup></p> |

The fields and options on this dialog are described below.

| Column/Button                                       | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also                                               |
|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <b>Search In</b>                                    | Select the type and name of each element field to search on.                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                        |
| <b>Condition</b>                                    | Select the condition of the search parameter. The available options are <b>Contains</b> , <b>Equal To</b> , <b>Not Equals</b> and <b>One Of</b> .                                                                                                                                                                                                                                                                                                                                                    | <a href="#">Fields and Conditions</a> <sup>[489]</sup> |
| <b>Look For</b>                                     | Specify the search term to perform the conditional search on. This value can pertain to the selected element field. For example, the value could be a date for <i>DateCreated</i> or a text value for other fields. The search term can contain multiple values, separated by commas.                                                                                                                                                                                                                | <a href="#">Fields and Conditions</a> <sup>[489]</sup> |
| <b>Required</b>                                     | <p>Indicate that the search results must include elements with your search term in that field. You select these checkboxes on the Add Filters dialog</p> <p>The fields listed as filters have an OR relationship when no <b>Required</b> checkboxes are selected; that is, if the search term is found in any one of those fields, then the element is displayed.</p> <p>Any field having the <b>Required</b> checkbox ticked overrides fields where the <b>Required</b> checkbox is not ticked.</p> | <a href="#">Adding Filters</a> <sup>[487]</sup>        |
| <b>Element Features</b><br>- Optional<br>- Required | <p>Specify whether element features are optional or required. These appear as a new branch underneath the root element term in the <b>Search In</b> column.</p> <p>If you scroll down the <b>Search In</b> column, you see sub-branches such as <i>Attribute</i>, <i>Change</i> and <i>Custom Property</i>. These are the element features.</p>                                                                                                                                                      |                                                        |

| Column/Button        | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                        |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
|                      | <p>You can add these features by clicking on the <b>Add Filter</b> button. The <b>Add Filters</b> dialog displays, with a list of all the filters you can choose for an element or element feature. Click on the <b>Search On</b> drop-down arrow to see a list of the element features you can search on. Each feature has its own set of filters such as <i>Name</i>, <i>Notes</i> and <i>Alias</i>, which you can add to your search. To search on an element attribute name, you would add the <i>Attribute</i> feature with a <i>Name</i> filter to your search.</p> <p>The <b>Optional</b> radio button enables you to generate a list of elements that meet <i>one</i> of the element features (<i>Element Type = Object</i>), or one of the feature filters (<i>Attribute Name = Class1</i>). For example, if your search is:</p> <p><i>Element Name = Class11, Attribute Name = m_At1 or Scope = Public</i></p> <p>the search results would list all the elements that have the <i>name</i> of <i>Class11</i> and all the elements that have an <i>Attribute Name</i> of <i>m_At1</i> or a <i>Scope</i> of <i>Public</i>.</p> <p>The <b>Required</b> radio button enables you to generate a list of elements that <i>must</i> have the element features you have added. For example, if your search is:</p> <p><i>Element Name = Class, Attribute Name = m_At1 or Scope = Public</i></p> <p>you would get elements that <i>must</i> have the <i>name</i> of <i>Class</i> AND an <i>Attribute</i> with a name of <i>m_att1</i> or a <i>Scope</i> of <i>Public</i>.</p> |                                                 |
| <b>Add Filter</b>    | Add a new set of parameters to filter the search on.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <a href="#">Adding Filters</a> <sup>[487]</sup> |
| <b>Edit Filter</b>   | Open the <b>Edit Filters</b> dialog, which enables you to change the search parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                 |
| <b>Remove Filter</b> | Remove the selected filter from the search.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                 |

### 17.1.1.10 Other Filters

The **Other Filters** tab enables you to define a set of filters to restrict your report to specific features of elements (sub-element components such as attributes, responsibilities or constraints). If the feature does not have the defined characteristics, it is not reported for the element.

#### Topics:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also                                                                                                                                                                      |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>How to Access</b> | <p>As with the <b>Document Options</b> tabs, you can access the filter details from two different places; the start point affects the persistence of the filter definition.</p> <ul style="list-style-type: none"> <li>If you define the filters on the Other Filters tab of the <b>Generate RTF Documentation</b> dialog, you can create filter settings for the current report to be run. Selections are non-persistent, and are reset when you close the dialog or select a different template.</li> <li>If you access the filter definitions by clicking on the <b>File   Document Options</b> menu option on the <b>RTF Style Template</b></li> </ul> | <p><a href="#">Generate RTF Documentation</a> <sup>[1732]</sup></p> <p><a href="#">File and Print Options</a> <sup>[1745]</sup></p> <p><a href="#">RTF Style Template</a></p> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See also                                                                                            |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
|       | <p><b>Editor</b> dialog, the settings are saved with the template as the default settings for any run of this report.</p> <p>You add filters by clicking on the <b>Add Filters</b> button. To edit the filters, either double-click on the panel contents or click on the Edit Filter button, to display the Edit Filters dialog. The format is the feature field name, the conditions placed on the field value, the actual value or delimiting value to search on, and whether the filter item is required (mandatory).</p> | <p><a href="#">Editor</a> <sup>[1736]</sup></p> <p><a href="#">Add Filters</a> <sup>[487]</sup></p> |

The fields and options on this dialog are described below.

**Reference:**

| Column/Button        | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                               |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <b>Search In</b>     | Select the name of each feature field to search on.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                        |
| <b>Condition</b>     | Select the condition of the search parameter. The available options are <b>Contains</b> , <b>Equal To</b> , <b>Not Equals</b> and <b>One Of</b> .                                                                                                                                                                                                                                                                                                                                                     | <a href="#">Fields and Conditions</a> <sup>[489]</sup> |
| <b>Look For</b>      | Specify the search term to perform the conditional search on. This value can pertain to the selected field. For example, the value could be a date for <i>DateCreated</i> or a text value for other fields. The search term can contain multiple values, separated by commas.                                                                                                                                                                                                                         | <a href="#">Fields and Conditions</a> <sup>[489]</sup> |
| <b>Required</b>      | <p>Indicate that the search results must include elements with your search term in that field. You select these checkboxes on the Add Filters dialog.</p> <p>The fields listed as filters have an OR relationship when no <b>Required</b> checkboxes are selected; that is, if the search term is found in any one of those fields, then the element is displayed.</p> <p>Any field having the <b>Required</b> checkbox ticked overrides fields where the <b>Required</b> checkbox is not ticked.</p> | <a href="#">Adding Filters</a> <sup>[487]</sup>        |
| <b>Add Filter</b>    | Add a new set of parameters to filter the search on.                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <a href="#">Adding Filters</a> <sup>[487]</sup>        |
| <b>Edit Filter</b>   | Open the <b>Edit Filters</b> dialog, which enables you to change the search parameters.                                                                                                                                                                                                                                                                                                                                                                                                               |                                                        |
| <b>Remove Filter</b> | Remove the selected filter from the search.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                        |

### 17.1.1.11 Project Constants

The **Project Constants** tab enables you to add values for your own project-specific fields in a report, to be inserted during compilation. The fields can be inserted into any template, in the text, headers or page headers and footers, using the RTF Style Template Editor **context menu**.

| Field      | Usage                                                                            | See also |
|------------|----------------------------------------------------------------------------------|----------|
| <b>Add</b> | Add a new field and value to the list. A prompt displays for the two data items. |          |

| Field  | Usage                                                                                                        | See also |
|--------|--------------------------------------------------------------------------------------------------------------|----------|
| Delete | Remove the selected field from the list. A prompt displays to confirm the deletion.                          |          |
| Import | Import a set of fields from an external XML file. A browser displays to select the source file and location. |          |
| Export | Export all fields to an external XML file. A browser displays to specify the target file and location.       |          |

Learn More:

- [Add Content](#)<sup>[1742]</sup>

### 17.1.1.12 Word Substitution

The **Word Substitution** tab of the **Generate RTF Documentation** dialog enables you to define translations of technical terms used in Enterprise Architect, in particular field names, into a language other than English for direct substitution into RTF documentation.

To add a translation for a term, follow the steps below:

How To:

| Step | Action                                                                                                                             | See Also |
|------|------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Double-click on the term in the <b>English</b> column in the <b>Word Substitution</b> List; the <b>Enter Value</b> field displays. |          |
| 2.   | Type the foreign language translation in the <b>Enter Value</b> field, and click on the <b>OK</b> button.                          |          |

### 17.1.1.13 Language Substitution

The **Codepage** tab of the **Generate RTF Documentation** dialog enables you to define languages other than English for direct substitution into RTF documentation.

If you export RTF-format documents from Enterprise Architect in languages other than English, you can customize the codepage, default language ID and character set that Enterprise Architect uses when generating RTF. This makes it much easier to generate documentation appropriate to your country or locale.

How To:

| Step | Action                                                                                                                                                                              | See Also |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | From the drop-down lists in the <b>Language</b> , <b>Codepage</b> and <b>Charset</b> fields, select the language, codepage and character set that most closely match your location. |          |
| 2.   | If required, modify the <b>Substitute Tag</b> by double-clicking on each and manually setting the value (for advanced use only).                                                    |          |
| 3.   | To clear the substitution list, double-click on each item in turn and delete                                                                                                        |          |

| Step | Action                | See Also |
|------|-----------------------|----------|
|      | the substitute value. |          |

Now when you generate RTF documents, the substitute tags are used in the output.

**Notes:**

- You can transport these tag definitions between models, using the **Export Reference Data** and **Import Reference Data** options on the Project | Model Import/Export menu

**Learn More:**

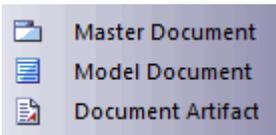
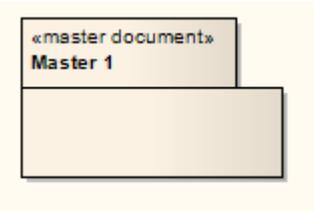
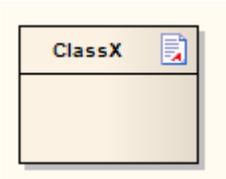
- [Export Reference Data](#)<sup>[237]</sup>
- [Import Reference Data](#)<sup>[239]</sup>

## 17.1.2 Virtual Documents

*Virtual documents* enable you to structure and filter your **RTF** and **HTML** reports by selecting, grouping and ordering individual packages independent of the organization of the Project Browser.

**Topics:**

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Abstract</b> | <p>You can create separate virtual documents defining, say, Requirements, Use Cases or Design elements of a project, or you can combine these separate reports - retaining their own different formats - into a single generated document with common headers and footers and a central contents list. This combined document could apply your corporate standards. You generate virtual documents in Enterprise Architect from individual <b>Model Document</b> elements. You can also, if required, combine several Model Documents under a <b>Master Document</b> package element. You can create as many Model Documents as required, for as many combinations of information as required.</p> <p>Each Model Document element has its own template; for example, a specifically-designed Requirements template for a Requirements document, or a Use Case template for a section on Use Cases. For HTML, the template is identified on the report generation dialog. For RTF, the template is identified in a Tagged Value.</p> <p>The content of the report is defined as either:</p> <ul style="list-style-type: none"> <li>A list of packages (defined as attributes) dragged onto the element in whatever order or combination is most appropriate to your requirements - you can easily <b>add</b> or <b>delete</b> packages as necessary; or</li> <li>(Not for HTML reports) a standard model search (defined as Tagged Values) created within the Model Search facility - note that diagram searches are not supported; when you generate the document, this search captures the required data throughout the model, and populates the document.</li> </ul> | <p><a href="#">Creating a Model Document</a><sup>[1777]</sup></p> <p><a href="#">Create a Master Document</a><sup>[1778]</sup></p> <p><a href="#">Add packages to your Document Object</a><sup>[1778]</sup></p> <p><a href="#">Deleting packages from a model document</a><sup>[1779]</sup></p> <p><a href="#">Create and Modify Searches</a><sup>[482]</sup></p> <p><a href="#">Import RTF Templates</a><sup>[1765]</sup></p> <p><a href="#">Document Order</a><sup>[1780]</sup></p> |

| Topic                           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also                                                 |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
|                                 | <p>The Master Document element has its own template Tagged Value, which defines the headers, footers and central contents list. You can <b>import your corporate standards template</b> and edit the Tagged Value to identify that. The template in the Master Document overrides the templates in the Model Documents. For example, headers and footers in the Master Document template override any header and footer definitions in the Model Document templates. This enables you to apply consistent and continuous styles and page numbering throughout the report generated through the Master Document. If you want the Model Documents to have their own styles, applied through their own RTFTemplate Tagged Values, either leave the Master Document RTFTemplate Tagged Value blank (for completely separate overall styles) or remove the definition of specific styles from the Master Document template.</p> <p>You can control the sequence in which information is presented in the document; see the <i>Document Order</i> topic.</p> |                                                          |
| <p><b>Document Elements</b></p> | <p>The Master Document and Model Document elements are available from the Documentation page of the Toolbox; on the Toolbox, select <b>More tools   Documentation</b>. (This Toolbox page also provides the <b>Document Artifact</b> element, which is not related to virtual documents but is used for adding a linked document to an element.)</p>  <p>When you drag the Master Document and Model Document elements onto a diagram, the following symbols display, respectively:</p> <div style="display: flex; justify-content: space-around;">   </div>                                                                                                                                                                                                                                 | <p><a href="#">Document Artifact</a> <sup>[95]</sup></p> |

**Notes:**

- In a Model Document for an RTF report, you should not define both a list of packages and a search; if both are present, when you generate the RTF document Enterprise Architect works from the package list only
- You cannot use **RTF Bookmarking** in Master Document elements, which effectively replace RTF Bookmarking in Word.

RTF Bookmarking requires each bookmark to be unique. When you generate a report with a standard RTF template (including in a single Model Document element), each bookmark is unique and there is a 1:1 association between the Elements-details being generated and the elements in the repository. As Master Documents are intended to contain multiple sub-documents, the association ceases to be 1:1. There is no simple method that enables the generated data to be uniquely identified directly in association with the original element

**Learn More:**

- [Model and Master Documents](#)<sup>[1813]</sup>
- [RTF Bookmarking](#)<sup>[1796]</sup>

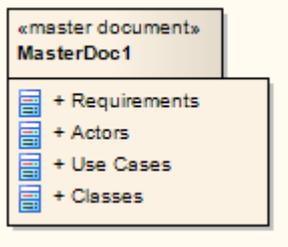
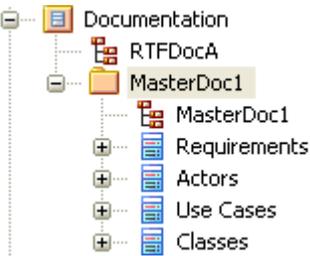
**17.1.2.1 Create Master Document**

Whilst you can create Model Document elements separately and generate individual documents from each one, you have added flexibility and scope if you organize Model Documents under a Master Document. You can generate a document with a **corporate template** for the covers, contents, headers and footers, whilst each section (generated from a separate Model Document) has its own appearance defined by a template appropriate to the section content.

**How To:**

To create a Master Document element, follow the steps below:

| Step | Action                                                                                                                                                                                                                                                                                                                                                                | See Also                                                |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| 1    | Open or create the diagram in which to create the Master Document                                                                                                                                                                                                                                                                                                     |                                                         |
| 2    | In the Toolbox, select <b>More tools   Documentation</b>                                                                                                                                                                                                                                                                                                              |                                                         |
| 3    | Drag the <i>Master Document</i> icon onto the diagram<br>The system prompts you for the name of the Master Document                                                                                                                                                                                                                                                   |                                                         |
| 4    | Type the element name and click on the <b>OK</b> button<br>The system creates the Master Document element and a child Custom diagram of the same name                                                                                                                                                                                                                 |                                                         |
| 5    | If creating a Master Document for an HTML report, go to step 7<br><br>Otherwise, open the Tagged Values window ( <b>View   Tagged Values</b> ) and click on the Master Document element<br><br>The <i>RTFTemplate</i> Tagged Value displays in the window. This Tagged Value is filled with the valid template names                                                  |                                                         |
| 6    | The RTFTemplate Tagged Value defaults to ( <i>model document: master template</i> )<br><br>If you want to use an alternative master template, click on the drop down arrow at the right of the field and click on that template in the list                                                                                                                           |                                                         |
| 7    | Return to the Project Browser and open the Master Document child diagram<br><br>At this point, you <b>create the Model Document</b> elements in the child diagram, to provide the content for the generated document<br><br>When you have added all your Model Document elements to the Master Document diagram, the Master Document element resembles the following: | <a href="#">Create Model Document</a> <sup>[1777]</sup> |

| Step | Action                                                                                                                                                                                                                                                                             | See Also |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      |  <p>Your completed Master document element and child diagram display in the Project Browser as shown below:</p>  |          |

**Learn More:**

- [Import RTF Template](#)<sup>[1765]</sup>

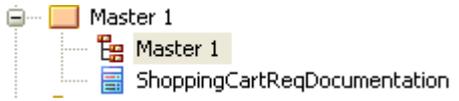
### 17.1.2.2 Create Model Document

You can create as many Model Document elements as are necessary to provide the sections of your generated document (under a Master Document) or to provide the independent documents you require.

**How To:**

To create a Model Document element, follow the steps below:

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                       | See Also                                                 |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| 1    | <p>Open the Master Document child diagram or (if you are creating independent Model Documents) create a new diagram</p> <p>The new diagram can live anywhere outside the packages you are adding to the document; you could create a Class diagram called Documentation within a specific Documentation package, and use this to hold the independent Model Document elements for your virtual documents</p> | <a href="#">Create Master Document</a> <sup>[1776]</sup> |
| 2    | <p>From the Documentation page of the Toolbox (<b>More tools   Documentation</b>) drag the <i>Model Document</i> icon onto the diagram to create a new Model Document element</p> <p>Give the element an appropriate name: for example, if the documentation is relevant to the shopping cart requirements of a model, you could call it <i>ShoppingCartReqDocumentation</i></p>                             |                                                          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                         | See Also                                                         |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
|      | Click on the <b>OK</b> button                                                                                                                                                                                                                                                                                                                                  |                                                                  |
| 3    | <p>If you are creating a Model Document element for an HTML report, go now to the <i>Add Packages to Model Document</i> topic</p> <p>Otherwise, open the Tagged Values window (<b>View   Tagged Values</b>) and click on the Model Document element</p> <p>The RTFTemplate, SearchName and SearchValue Tagged Values display in the window</p>                 | <a href="#">Add Packages to Model Document</a> <sup>[1778]</sup> |
| 4    | Click on the drop-down arrow to the right of the <b>RTFTemplate</b> field, and click on the template to use for this Model Document                                                                                                                                                                                                                            |                                                                  |
| 5    | <p>If you are creating a list of packages for the Model Document, go now to the <i>Add Packages to Model Document</i> topic</p> <p>Otherwise, click on the drop-down arrow to the right of the <b>SearchName</b> field, and click on the model search type to populate this Model Document</p>                                                                 | <a href="#">Add Packages to Model Document</a> <sup>[1778]</sup> |
| 6    | If necessary, type a search term in the <b>SearchValue</b> field                                                                                                                                                                                                                                                                                               |                                                                  |
| 7    | <p>Create further Model Document elements as required.</p> <p>Your Model Document element appears in the Project Browser with a Class icon, as shown below:</p>  <p>When you have created all the required Model Document elements, see the <i>Document Order</i> topic</p> | <a href="#">Document Order</a> <sup>[1780]</sup>                 |

**Notes:**

- Diagram Searches are not supported
- Custom SQL searches are supported if they are returning elements; the SQL must include *ea\_guid* AS *CLASSGUID* and the *object type*

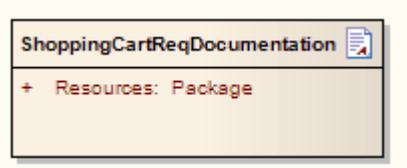
**Learn More:**

- [Creating Search Definitions](#)<sup>[484]</sup>

**17.1.2.3 Add Packages to Model Document****How To:**

To add packages to your Model Document element, follow the steps below

(As the example Model Document element here is called *ShoppingCartDocumentation*, the steps indicate how to add shopping cart-related packages to the element.)

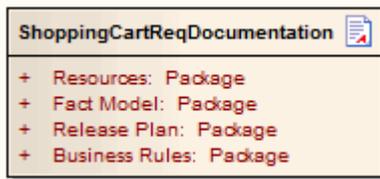
| Step | Action                                                                                                                                                                                                                                                                                                         | See Also                                                                                                                                                                                                                                                                          |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Keeping the documentation diagram open, find a package in the Project Browser to add to the documentation; for example, a Resources package in a Dynamic view                                                                                                                                                  |                                                                                                                                                                                                                                                                                   |
| 2    | <p>Drag and drop the package from the Project Browser onto the Model Document element, as shown below:</p>                                                                                                                   |                                                                                                                                                                                                                                                                                   |
| 3    | <p>The title of the package displays in the Model Document element in the <i>Attributes</i> compartment, as shown below:</p>  <p>This means that the Resources package is included in the document when you generate it</p> |                                                                                                                                                                                                                                                                                   |
| 4    | <p>Using the above method, you can add as many packages from as many different views as required</p> <p>The next step is to generate your document in RTF or HTML, but consider the impact of your package list on the Document Order; you can also delete packages if required</p>                            | <p><a href="#">Generating RTF Documents</a><sup>[1782]</sup></p> <p><a href="#">Generating a Master Document - HTML</a><sup>[1814]</sup></p> <p><a href="#">Document Order</a><sup>[1780]</sup></p> <p><a href="#">Delete Packages from a Model Document</a><sup>[1779]</sup></p> |

#### 17.1.2.4 Delete Package in Model Document

You can delete a package from your Model Document element.

This example includes four packages:

- Resources Package
- Activity Diagram Package
- View Cart Package
- Sequence Diagram Package

**How To:**

To delete a package from a Model Document element, follow the steps below:

| Step | Action                                                                                           | See Also |
|------|--------------------------------------------------------------------------------------------------|----------|
| 1    | In the Project Browser, expand the element to list the package attributes                        |          |
| 2    | Right-click on the package to delete, and select the <b>Delete Attribute</b> context menu option |          |

Alternatively:

| Step | Action                                                                                                                                                                         | See Also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | In either the Project Browser or the diagram, right-click on the Model Document element and select the <b>Attributes</b> context menu option<br>The Attributes dialog displays |          |
| 2    | On the <b>Attributes</b> list, click on the package to delete                                                                                                                  |          |
| 3    | Click on the <b>Delete</b> button to remove the package from the document element                                                                                              |          |

**17.1.2.5 Document Order**

The order in which information is compiled into a document depends on:

- The sequence of Model Document elements in a Master Document element
- Whether you define a Model Search in a Model Document element (not for HTML reports)
- Whether you define a package list in a Model Document element

When you have considered and, if necessary, amended the order in which information is compiled, you can generate the document in RTF or HTML.

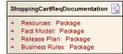
**Topics:**

| Topic                          | Detail                                                                                                                                                                                                                | See also |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Model Document Sequence</b> | When you generate a document from a Master Document element, the sequence in which the sections are generated is determined by the order in which the child Model Document elements are listed in the Project Browser |          |

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                      | <p>You can create elements anywhere in a diagram, therefore the generator refers to the Project Browser sequence</p> <p>If necessary, change the sequence using the green Up and Down arrows in the Project Browser toolbar to move an element up or down within the package</p>                                                                                                                                                                         |          |
| <b>Model Search</b>  | <p>A Model Search operates on the database, and accesses records in the order in which they are stored; this order depends on many factors, and can change with database maintenance</p> <p>Therefore, the sequence of information provided by the search is unpredictable</p>                                                                                                                                                                           |          |
| <b>Package Order</b> | <p>When you create a package list in a Model Document element, the sequence of information is determined by the order in which the package attributes are listed within the element</p> <p>You can change the sequence using the Attributes dialog, and if you prefer a package to be in a different section of the document, you can move the attribute from one Model Document element to another</p> <p>Both these procedures are described below</p> |          |

### How To:

To change the sequence of Packages in a Model Document element, refer to the example element and follow the steps below:

| Images                                                                              | Step | Action                                                                                                                                                                                  | See Also |
|-------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|  | 1    | <p>Right-click on the Model Document and select the <b>Attributes</b> option from the context menu</p> <p>The Attributes dialog displays</p>                                            |          |
|                                                                                     | 2    | <p>On the <b>Attributes</b> list, click on a package to move and click on the Up or Down (hand) buttons to change the order in which the packages are included in the documentation</p> |          |
|                                                                                     | 3    | <p>When you are satisfied with the order of your packages, click on the <b>Close</b> button</p>                                                                                         |          |

To move a package attribute from one Model Document to another, follow the steps below

| Step | Action                                                                                                                      | See Also |
|------|-----------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | <p>Expand the Model Document elements in the Project Browser, so that both list their package attributes</p>                |          |
| 2    | <p>Click-and-hold on the attribute to move, and drag it onto the name of the target Model Document element</p>              |          |
| 3    | <p>Release the mouse button</p> <p>The attribute is removed from the source element and added to the top of the list of</p> |          |

| Step | Action                                                                                                                       | See Also |
|------|------------------------------------------------------------------------------------------------------------------------------|----------|
|      | attributes in the target element                                                                                             |          |
| 4    | If necessary, move the attribute down the attribute list, as described in <i>Change Sequence of Packages in List</i> , above |          |

**Learn More:**

- [Generate the Document \(RTF\)](#) <sup>[1782]</sup>
- [Generate a Master Document HTML](#) <sup>[1814]</sup>

### 17.1.2.6 Section Numbering in Virtual Documents

Virtual documents can contain several sections based on separate templates. To ensure that section and subsection numbering continues sequentially through the sections, you can either use:

- The *MasterList* style from the **Normal.rtf** file (which is the default applied to all new templates, such as those used in each Master Document and Model Document element); this provides a simple but consistent list numbering style that you can start off with. Alternatively, use:
- Your own numbering style

If you design your own styles in the template used in each model document, each template must use the same **List Overrides** for section numbering. These have a unique ID, so to propagate the List Overrides across multiple templates, you must:

| Step | Action                                                                                                                                                                                      | See Also                                                 |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| 1    | Set up the common List Overrides in the <i>Normal.rtf</i> template (your own customized override list)                                                                                      |                                                          |
| 2    | Within each of the other templates used in the virtual document, using the template editor, select <b>File   Update Styles</b> to update the templates to include the common List Overrides | <a href="#">File and Print Options</a> <sup>[1745]</sup> |
| 3    | In each template use this common List Level for progressive numbering throughout the virtual document                                                                                       |                                                          |

**Learn More:**

- [RTF Templates Tab](#) <sup>[1735]</sup>
- [User Defined Section Numbering](#) <sup>[1761]</sup>

### 17.1.2.7 Generate the Document (RTF)

**How To:**

To generate the documentation defined in the Master Document and/or Model Documents, follow the steps below

| Step | Action                                                                                                           | See Also |
|------|------------------------------------------------------------------------------------------------------------------|----------|
| 1    | On the documentation diagram, click on the Master Document element (or on an independent Model Document element) |          |

| Step | Action                                                                                                                                                             | See Also                                                            |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| 2    | Select the <b>Element   Advanced   Rich Text Format ( RTF) Report</b> menu option<br>The Generate RTF Documentation dialog displays                                |                                                                     |
| 3    | Set the options for your RTF document as required<br>See the <i>Generate RTF Documentation Dialog</i> and related topics for further information on these settings | <a href="#">Generate RTF Documentation Dialog</a> <sup>[1732]</sup> |
| 4    | Click on the <b>Generate</b> button to create the documentation                                                                                                    |                                                                     |
| 5    | If you have not selected the <b>View Document on Completion</b> checkbox, click on the <b>View</b> button to view the documentation                                |                                                                     |

The RTF Report Generator works through the defined content of the Master Document element and/or the Model Document elements and pulls in the information from either the listed packages or the executed searches, formatted according to the templates identified in the *RTFTemplate* Tagged Value for each document element.

### 17.1.3 Other Documents

Enterprise Architect has other RTF based documentation that you can output:

| Topic                                                                                                                                                                                              | Link                                                     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| <b>Dependency Report</b> - shows a list of any elements that are dependent (in a Dependency connector) on another element for their specification                                                  | <a href="#">Dependency Report</a> <sup>[1783]</sup>      |
| <b>Diagrams Only Report</b> - lists only the relevant diagrams from the target package                                                                                                             | <a href="#">Diagrams Only Report</a> <sup>[1784]</sup>   |
| <b>Implementation Report</b> - lists, for a specified package, the elements that require implementers, together with any source elements in Realize (Implements) relationships with those elements | <a href="#">Implementation Report</a> <sup>[1785]</sup>  |
| <b>Resource Report</b> - shows how your resources are deployed in your project, displaying a list of all elements that have resources allocated to them                                            | <a href="#">Resource Report</a> <sup>[370]</sup>         |
| <b>Testing Report</b> -                                                                                                                                                                            | <a href="#">Testing Report</a> <sup>[1786]</sup>         |
| <b>Testing Details Report</b> - enables you to review the testing details for a package, as filtered reports on all elements in the package hierarchy under the selected package                   | <a href="#">Testing Details Report</a> <sup>[1786]</sup> |

#### 17.1.3.1 Dependency Report

A Dependency report shows a list of any elements in the package (selected from the Project Browser) that are dependent on another element for their specification.

For example, a Use Case derives its specification from the Requirement that it realizes.

Each of the elements in the first column of the report is the source or dependent in a Dependency connector to the corresponding target element in the **Dependent on** column.

**Access:** [Project | Documentation | Dependency Details](#)

**How to:**

On the Dependency Details dialog, run the report and make selections from the options as indicated below:

| Field                | Usage                                                                                                                                                                                                                                | See also                            |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| <b>Root Package</b>  | Confirm the root package; all elements and packages under this package appear in the report<br><br>If you want to change the package, click on the replacement package in the Project Browser and click on the <b>Refresh</b> button |                                     |
| <b>Refresh</b>       | Run the report                                                                                                                                                                                                                       |                                     |
| <b>Locate Object</b> | Locate the selected element in the Project Browser                                                                                                                                                                                   |                                     |
| <b>Print</b>         | Print the list                                                                                                                                                                                                                       |                                     |
| <b>Save Report</b>   | Display a small dialog in which you specify the file location into which to save the report as an RTF file                                                                                                                           |                                     |
| <b>Details</b>       | List dependency details; lists the elements in the current package and the elements that they are dependent on                                                                                                                       | <a href="#">Dependency</a><br>[979] |

### 17.1.3.2 Diagrams Only Report

You can also produce an RTF report that contains only the relevant diagrams from the target package. This is convenient for printing or handling a lot of diagrams in batch, rather than exporting or printing each one at a time.

**Access:** [Project Browser package context menu](#) | [Documentation](#) | [Diagrams Only Report](#)

**How To:**

To produce a Diagrams Only report, follow the steps below:

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | On the Export Diagrams to RTF Document dialog, in the <b>Output Path</b> field, type or browse for the output location to create the report in                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |
| 2    | Select the options you require, as follows: <ul style="list-style-type: none"> <li>• Select the <b>Embed Diagrams in Document</b> checkbox to ensure the diagrams are created within the RTF file, not as linked image files</li> <li>• Select the <b>Include all child packages</b> checkbox to document all of the diagrams included in any child package</li> <li>• Select the <b>Include Diagram Name</b> checkbox to include the diagram name within the generated documentation</li> <li>• Select the <b>Order Diagrams Alphabetically</b> checkbox to generate the documentation in alphabetical order</li> </ul> |          |
| 3    | Click on the <b>Generate</b> button to run the report                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |          |
| 4    | When the report is generated, click on the <b>View</b> button to show the RTF output                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |          |

### 17.1.3.3 Implementation Report

An *Implementation report* shows a list of the elements in the package (selected in the Project Browser) that require implementers, together with any source elements in Realize (Implements) relationships with those elements.

**Access:** [Project](#) | [Documentation](#) | [Implementation Details](#)

#### How to:

On the Implementation Details dialog, make selections from the options as indicated below, and run the report:

| Field                     | Usage                                                                                                                                                                                                                                                          | See also                                                        |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| <b>Root Package</b>       | Confirm the root package; all elements and packages under this package appear in the report<br><br>If you want to report on a different package, click on that package in the Project Browser                                                                  |                                                                 |
| <b>Set Target Types</b>   | By default Enterprise Architect only reports on a limited number of element types, such as Use Cases and Requirements; you can expand the range of element types to include<br><br>Set the list of types to report on, using the Implementation Targets dialog | <a href="#">Implementation Targets Dialog</a> <sup>[1785]</sup> |
| <b>Refresh List</b>       | Run the report and display the results                                                                                                                                                                                                                         |                                                                 |
| <b>Locate Object</b>      | Locate the selected element in the Project Browser                                                                                                                                                                                                             |                                                                 |
| <b>Show Unimplemented</b> | Show non-implemented elements<br><br>Non-implemented elements are those that don't have any other element to realize them (for example, a Use Case that has no Component or Class to implement the Use Case behavior)                                          |                                                                 |
| <b>Show Implemented</b>   | Show implemented elements<br><br>These are elements that do have some element associated with them in a Realize relationship (for example a Use Case that has a Component that implements it)                                                                  | <a href="#">Realize</a> <sup>[1007]</sup>                       |
| <b>Details</b>            | List elements in the current package hierarchy and elements that implement them                                                                                                                                                                                |                                                                 |
| <b>Print</b>              | Print the list                                                                                                                                                                                                                                                 |                                                                 |
| <b>Save Report</b>        | Specify the file location into which to save the report as an RTF file                                                                                                                                                                                         |                                                                 |

#### 17.1.3.3.1 Implementation Targets Dialog

The Implementation Targets dialog is accessed by clicking on the **Set Target Types** button on the Implementation Details dialog.

This dialog enables you to set the types of elements that appear in the report as requiring implementation.

Double-click on an element in either list to move it to the other list.

### 17.1.3.4 Testing Report

**Access:** [Project | Documentation | Testing Report](#)

**How To:**

To view a Testing report, follow the steps below:

| Step | Action                                                                                                                      | See Also |
|------|-----------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | In the Project Browser, click on the package to report on<br>The package name displays in the <b>Root Package</b> field     |          |
| 2    | In the Report on panel, select the check box against each type of test to include in the report                             |          |
| 3    | If you intend to report on tests on elements in the subordinate packages, select the <b>Include child packages</b> checkbox |          |
| 4    | In the Output file field, type in or browse for the file path of the report output file                                     |          |
| 5    | Click on the <b>Generate</b> button to output the report, and open it at the specified file location                        |          |

### 17.1.3.5 Testing Details Report

The Testing Details report enables you to review the testing details for a package, running filtered reports on all elements in the package hierarchy under the selected package.

**Access:** [Project | Documentation | Testing Details](#)

**Use to:**

- Show the testing details for the elements in a package, for tests that meet specific criteria

**How to:**

On the Testing tab, make selections from the options as indicated below, and run the report:

| Field             | Usage                                                                                                     | See also |
|-------------------|-----------------------------------------------------------------------------------------------------------|----------|
| <b>Run By</b>     | Select a name to filter for tests run by that person; click on the <b>x</b> button to clear the field     |          |
| <b>Checked By</b> | Select a name to filter for tests checked by that person. Click on the <b>x</b> button to clear the field |          |
| <b>Test Type</b>  | Select the radio button for the required test type                                                        |          |
| <b>Status</b>     | Select the radio button for the required test status                                                      |          |

| Field                | Usage                                                                                             | See also |
|----------------------|---------------------------------------------------------------------------------------------------|----------|
| <b>Locate Object</b> | (After clicking on an element in the Test Details list) locate the element in the Project Browser |          |
| <b>Refresh</b>       | Re-run the report query                                                                           |          |
| <b>Print</b>         | Print a summary of the test results                                                               |          |

**Notes:**

- If required, you can display the test details on the Testing window, by double-clicking on the test line in the report

**Learn More:**

- [The Testing Workspace](#)<sup>[1695]</sup>

### 17.1.4 The Legacy RTF Report Generator

**Topics:**

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Abstract</b> | <p>Creating a <b>Rich Text Format</b> (RTF) document is a simple and flexible process. An RTF document is based on a package or an element in your project (more usually a package). To produce a document, you select the package or element to report on in the <b>Project Browser</b>, <b>Package Browser</b>, <b>Diagram List</b> or <b>Model Search</b>, then press ( <b>F8</b> ) to display the <b>Generate RTF Documentation</b> dialog. On the <b>Advanced</b> tab, click on the <b>Switch generator</b> button to access the Legacy <b>Rich Text Format Report</b> dialog.</p> <p>The <b>Rich Text Format Report</b> dialog enables you to set the exact contents and look and feel of your report. You enter the file name of the report, a heading, additional notes, template name (for saving the set-up) and other options. You can also select the style of the report; either plain or formal.</p> <p>Optionally, you can set up a filter, the details to include, element types to exclude, whether to process child packages, whether to show diagrams and the diagram format.</p> <p>You can switch back to the <b>Generate RTF Documentation</b> dialog by clicking on the <b>Switch RTF Generator</b> button.</p> |          |

**Notes:**

- The Legacy Generator is available if you have RTF templates created in releases of Enterprise Architect prior to 4.1, and you prefer to generate RTF reports using the original generator. However, as you can generate reports from these templates using the post-Enterprise Architect 4.1 RTF Generator, the Legacy Generator and instructions for its use are no longer updated. However, reports produced using the Legacy RTF Generator do reflect the Notes formatting feature in any text associated with elements
- The **Rich Text Format Report** dialog panels are individually described in the subsequent topics of this section (listed below). The dialog has a lot of options; get to know them all to produce output at the level of detail suited to your project

**Learn More:**

- [Document a Single Element](#)<sup>[1788]</sup>
- [Set the Main RTF Properties](#)<sup>[1788]</sup>
- [Apply a Filter](#)<sup>[1789]</sup>
- [Exclude Elements](#)<sup>[1790]</sup>
- [RTF Diagram Format](#)<sup>[1790]</sup>
- [Model Include](#)<sup>[1790]</sup>
- [RTF Report Options](#)<sup>[1790]</sup>
- [RTF Report Selections](#)<sup>[1791]</sup>
- [Generate the Report](#)<sup>[1792]</sup>
- [Diagrams Only Report](#)<sup>[1784]</sup>
- [Report Templates](#)<sup>[1792]</sup>
- [Include or Exclude a Package from Report](#)<sup>[1731]</sup>
- [Save as Document](#)<sup>[1793]</sup>
- [Generate RTF Documentation Dialog](#)<sup>[1732]</sup>
- [Notes Toolbar](#)<sup>[770]</sup>

**17.1.4.1 Document a Single Element**

RTF documentation can also be generated for a single element.

**Access:** **Select an element and then go to: Element | Advanced | Rich Text Format (RTF) Report**

Click on the **Switch Generator** button to display the **Rich Text Format Report** dialog. See The **Legacy RTF Report Generator** and its related topics for further information.

**Learn More:**

- [Generate RTF Documentation Dialog](#)<sup>[1732]</sup>
- [The Legacy RTF Report Generator](#)<sup>[1787]</sup>

**17.1.4.2 Set the Main RTF Properties**

The main section of the **Rich Text Format Report** dialog enables you to set the output location and appearance of the final RTF document.

**How To:**

| Step | Action                                                                                                                | See Also                                                          |
|------|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 1.   | Open the <b>Rich Text Format Report</b> dialog (see The <b>Legacy RTF Report Generator</b> topic for how to do this). | <a href="#">The Legacy RTF Report Generator</a> <sup>[1787]</sup> |
| 2.   | Supply an <b>Output Filename</b> to save the report into; always include the extension .RTF as part of the filename.  |                                                                   |
| 3.   | Provide a <b>Template Name</b> to save this report set-up.                                                            |                                                                   |
| 4.   | Select a report <b>Style</b> : Formal or Basic.                                                                       |                                                                   |
| 5.   | Type a <b>Heading</b> for your report; this appears as the first heading item in your output.                         |                                                                   |

| Step | Action                                                                                               | See Also |
|------|------------------------------------------------------------------------------------------------------|----------|
| 6.   | Select your required <b>Heading Style and Initial Heading Level Indent</b> from the drop-down lists. |          |

**Notes:**

- It is recommended that you enter a full path name for your report. The images in your report are saved externally in an images directory, and supplying the full directory path avoids confusion over the location of these images. Also, if you move your report you must also move the images directory

**17.1.4.3 Apply a Filter****Topics:**

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Usage | <p>You can apply a filter on the <b>Rich Text Format Report</b> dialog to include or exclude elements by date modified, phase or status. This helps to track changes and break a document into multiple delivery phases.</p> <p>Open the <b>Rich Text Format Report</b> dialog (see <b>The Legacy RTF Report Generator</b> for how to do this).</p> <ul style="list-style-type: none"> <li>• To enable the date filter, select the checkbox in the date field.</li> <li>• In the first two <b>Only include objects</b> fields, click on the drop-down arrows and select the appropriate criteria (<b>Modified/ Created, Before/After</b>).</li> <li>• The package phase filter applies at the package level (not the element level) and ignores the phase of the root package that you are documenting. To enable the phase filter, in the <b>Where Package Phase</b> field click on the drop-down arrow and select an operator; Enterprise Architect filters out all packages that do not meet the selection criteria. All elements in the package are ignored, regardless of their individual phase.</li> <li>• The element status filter enables you to limit the output by element status. Unlike the package phase filter, this filter applies to every element. You can filter against a status of like or <i>unlike</i> a criterion, for example, <i>like proposed</i>, or against the <i>in</i> and <i>not in</i> operators, such as <i>in approved</i>, <i>not in validated</i>. When using the <i>in</i> and <i>not in</i> operators, enter a comma-separated list of status types as your criteria expression</li> </ul> |          |

**Learn More:**

- [The Legacy RTF Report Generator](#)<sup>[1787]</sup>

### 17.1.4.4 Exclude Elements

#### Topics:

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See also                                                          |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| Usage | <p>The <b>Rich Text Format Report</b> dialog enables you to exclude elements of any type from your final output. This is useful when you want to highlight particular items and not clutter up a report with too much detail.</p> <p>Open the <b>Rich Text Format Report</b> dialog (see <b>The Legacy RTF Report Generator</b> for how to do this).</p> <p>Click on each element to exclude, or click on the <b>All</b> button to exclude all elements. Click on the <b>None</b> button to clear your selections.</p> | <a href="#">The Legacy RTF Report Generator</a> <sup>[1787]</sup> |

### 17.1.4.5 RTF Diagram Format

#### Topics:

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also                                                          |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| Usage | <p>You can output diagrams to Bitmap files, GIF files or Windows Metafiles.</p> <p>Open the <b>Rich Text Format Report</b> dialog (see <b>The Legacy RTF Report Generator</b> for how to do this). In the <b>Diagram format</b> panel (bottom center of the dialog) select the required format for the report. Generally the two metafile options (EMF and WMF) are recommended, however there are times that the others might be suitable.</p> | <a href="#">The Legacy RTF Report Generator</a> <sup>[1787]</sup> |

### 17.1.4.6 Model Include

#### Topics:

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                           | See also                                                                                                                                               |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Usage | <p>The <b>Model Include</b> panel of the <b>Rich Text Format Report</b> dialog has the following options:</p> <ul style="list-style-type: none"> <li>• Glossary to include the project glossary</li> <li>• Tasks to include project tasks</li> <li>• Issues to include project issues</li> </ul> <p>Select the appropriate checkbox to include the items in the generated RTF documentation.</p> | <a href="#">Project Glossary</a> <sup>[362]</sup><br><a href="#">Project Tasks</a> <sup>[356]</sup><br><a href="#">Project Issues</a> <sup>[358]</sup> |

### 17.1.4.7 RTF Report Options

Additional RTF report options you can select from the **Options** panel on the **Rich Text Format Report** dialog are shown below.

#### Topics:

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See also                                                                 |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| Usage | <p>You can select whether or not to recursively document packages, show diagrams or add a page break before each new package. Select the:</p> <ul style="list-style-type: none"> <li>• <b>Process all Children</b> checkbox to recursively process all child packages within the main package</li> <li>• <b>Show Diagrams</b> checkbox to include diagrams in your document. Clear this item for no diagrams</li> <li>• <b>New page per package</b> checkbox to force a page break on each new package (excepting empty packages)</li> <li>• <b>Document all elements</b> checkbox to include all elements included in the project</li> <li>• <b>Document Packages</b> checkbox to document the package as an element in addition to the documentation that would normally be produced for package documentation</li> <li>• <b>Hide 'note-less' elements</b> checkbox to exclude all elements without notes from the documentation</li> <li>• <b>Embed Diagrams in Document</b> checkbox to ensure that the diagram images are contained within the RTF document rather than stored in a linked external file</li> <li>• <b>Skip root package</b> checkbox to exclude the parent package from the documentation and include only the child packages</li> <li>• <b>Document Linked Elements</b> checkbox to include the object details for linked elements that do not originate from the selected package</li> <li>• <b>Use Heading styles for Details</b> checkbox to ensure that the details are formatted as heading styles rather than formatted text; this option is only available when the Heading Style field in the <b>Main section</b> of the <b>Rich Text Format Report</b> dialog is set to Max 9 levels - elements are package + 1.</li> </ul> | <a href="#">Setting the Main RTF Properties</a><br><small>[1788]</small> |

#### 17.1.4.8 RTF Report Selections

The **For each Object Include** section of the **Rich Text Format Report** dialog enables you to select the documentation sections to include in your report.

##### Topics:

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Usage | <p>What you include or exclude governs how simple or detailed your report is. You can create multiple reports at different levels of detail for different audiences. Experiment with these options to see what effect inclusion or exclusion has. Most items are self-explanatory.</p> <p>Selecting the checkbox against a category item in the list selects all of the options that are contained in the category. To expand a category, click on the +symbol next to the category name. To exercise greater control over a category of options expand the top level item and then select the required individual items from the list.</p> <p>Sometimes an item applies only to a certain type of element; for example, <b>Attributes</b> only applies to Class elements and a few other element types. The <b>Child Diagrams</b> option shows or hides any diagrams that are attached under a model element; for example, a Use Case might have a Scenario diagram attached.</p> |          |

**Notes:**

- Use this feature to produce the right level of detail for your audience. Technical readers might want to see everything, whilst management might require only the general outline

**17.1.4.9 Generate the Report****Topics:**

| Topic        | Detail                                                                                                                                                                                                                            | See also |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Usage</b> | Once you have set up the document properties as required, click on the <b>Generate</b> button to generate the report.<br><br>When you have generated the document, click on the <b>View</b> button to open the report in MS Word. |          |

**17.1.4.10 Legacy RTF Style Templates**

The **Legacy RTF Style Editor** enables you to edit the RTF associated with various sections of the RTF Report facility in Enterprise Architect. You would typically use this functionality to customize a report's look and feel for your company or client.

If you have previously defined and saved a template, click on the **Load** button on the **Rich Text Format Report dialog** to open the list of defined templates. Select one in order to load it as the current template; all the features saved become the current features. This enables you to define a set of standard report types that streamline document production.

**Access:** [View | More Project Tools | Project Resources | Resources](#)

To create or Edit RTF Style Templates, follow the steps below

**How To:**

| Step | Action                                                                                                                                                                                                                                              | See Also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Select the <b>View   More Project Tools   Project Resources</b> menu option to display the <b>Resources</b> window.                                                                                                                                 |          |
| 2.   | Expand the <b>Templates</b> folder                                                                                                                                                                                                                  |          |
| 3.   | To edit an <i>existing</i> Legacy template, expand the <b>Legacy Templates</b> tree and double-click on the template name, or right-click and select the <b>Modify Document Template</b> context menu option. The <b>RTF Style Editor</b> displays. |          |
| 4.   | Alternatively, to create a <b>new Legacy template</b> , right-click on <b>Legacy Templates</b> and select the <b>Create RTF Style Template (Legacy)</b> context menu option. Enterprise Architect displays a prompt for the new template name.      |          |
| 5.   | Type the name of the new template and click on the OK button. The <b>RTF Style Editor</b> displays.                                                                                                                                                 |          |

**Topics:**

| Topic            | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See also                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| RTF Style Editor | <p>The <b>RTF Style Editor</b> contains a list of all available RTF fragments for modification and customization.</p> <p>Each fragment typically contains RTF plus one or more special tag names that Enterprise Architect replaces with information during generation. Currently you cannot alter the content within the tag names, but you can omit a complete tag by removing it, or alter its basic display properties in the surrounding RTF.</p> <p>Special tag names are delimited by # characters; for example, <b>#NOTES#</b></p> <p>Click on the:</p> <ul style="list-style-type: none"> <li>• <b>Get Default</b> button to retrieve the default Enterprise Architect template for the currently-selected template item in the left hand list</li> <li>• <b>Save</b> button to save the version of the template for this style only</li> <li>• <b>Delete</b> button to remove your modified version of the template, which causes Enterprise Architect to use the default template during report generation</li> </ul> <p>To select a template during report generation, click on the <b>Style</b> drop-down arrow on the <b>Rich Text Format Report</b> dialog. Once a style is selected, Enterprise Architect applies that to the current report. Select &lt;Basic&gt; for the inbuilt style.</p> | <a href="#">The Legacy RTF Report Generator</a><br><small>1787</small> |

**Notes:**

- The RTF Style Editor discussed here automatically displays when you modify or create a Legacy RTF template. If you select a template created in the enhanced **RTF Style Template Editor**, that editor opens automatically instead
- You can transport these RTF templates between models, using the **Export Reference Data** and **Import Reference Data** options on the **Project | Model Import/Export** menu
- To delete a template, right-click on it and select the **Delete Document Template** context menu option
- You can also **alter the custom language settings**

**Learn More:**

- [RTF Style Template Editor](#)1736
- [Export Reference Data](#)237
- [Import Reference Data](#)239
- [Custom Language Settings](#)1794

**17.1.4.11 Save as Document**

The *Document* feature enables a particular documentation configuration to be 'remembered', linking the loaded template within the **Rich Text Format Report** dialog to the current highlighted package. If a particular template is always used with a specific package, and multiple cases of documentation exist to be propagated, saving these as Documents can ease document generation later.

To create and use Documents, follow the steps below

**How To:**

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See Also                                                          |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 1.   | Open the <b>Rich Text Format Report</b> dialog (see <b>The Legacy RTF Report Generator</b> for instructions on how to do this).                                                                                                                                                                                                                                                                                                                                                                                                                                                    | <a href="#">The Legacy RTF Report Generator</a> <sup>[1787]</sup> |
| 2.   | Click on the <b>Save as Document</b> button. The Save current as document definition dialog displays:                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                   |
| 3.   | In the <b>Enter Value</b> field, type a name for the document and click on the <b>OK</b> button. The document is added to the Resources window for easy future access                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                   |
| 4.   | To generate documentation from the <b>Resources</b> window, right-click on the required document. The context menu displays.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                   |
| 5.   | Select the required option.<br>The context menu options are: <ul style="list-style-type: none"> <li>• <b>Open Document</b> - Opens the corresponding .RTF file, as specified by the RTF template Filename property</li> <li>• <b>Generate Document</b> - Opens the <b>Rich Text Format Report</b> dialog, loaded with the specified template</li> <li>• <b>Auto Generate Document</b> - Generates documentation, with the document located at the path specified by the template's Filename property</li> <li>• <b>Delete Document</b> - Removes the specified document</li> </ul> |                                                                   |

#### 17.1.4.12 Custom Language Settings

If you export RTF-format documents from Enterprise Architect in languages other than English, you can customize the codepage, default language ID and character set that Enterprise Architect uses when generating RTF. This makes it much easier to generate documentation appropriate to your country or locale.

You can also set up a list of word substitutions. For instance, where Enterprise Architect would include the word *Figure*, you can specify another word to replace it that is either in your language or more meaningful to your readers.

To Set Up Substitutions, follow the steps below

##### How To:

| Step | Action                                                                                                                              | See Also                                                          |
|------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 1.   | Open the <b>Rich Text Format Report</b> dialog (see <b>The Legacy RTF Report Generator</b> for how to do this).                     | <a href="#">The Legacy RTF Report Generator</a> <sup>[1787]</sup> |
| 2.   | In the <b>Language</b> panel (bottom left of dialog) click on the Adjust button. The <b>Customize RTF Language</b> dialog displays. |                                                                   |
| 3.   | Double-click on an item to set or clear its <b>Substitute</b> word.                                                                 |                                                                   |
| 4.   | When you have finished, click on the <b>OK</b> button.                                                                              |                                                                   |

To Set Up Codepage and Character Set, Follow the steps below

##### How To:

| Step | Action                                                                                                                                                                              | See Also |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | From the drop-down lists in the <b>Language</b> , <b>Codepage</b> and <b>Charset</b> fields, select the language, codepage and character set that most closely match your location. |          |
| 2.   | If required, modify the <b>Substitute Tags</b> by double-clicking on each and manually setting the value (for advanced use only).                                                   |          |
| 3.   | To clear the substitution list, double-click on each item in turn and delete the substitute value.                                                                                  |          |
| 4.   | When you have completed the settings, click on the <b>OK</b> button to save them.                                                                                                   |          |

Now when you generate RTF documents, the substitute tags are used in the output.

**Notes:**

- You can transport these language and tag definitions between models, using the **Export Reference Data** and **Import Reference Data** options on the **Project | Model Import/Export** menu

**Learn More:**

- [Export Reference Data](#)<sup>[237]</sup>
- [Import Reference Data](#)<sup>[239]</sup>

## 17.1.5 Use MS Word

**Topics:**

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also                                                                                                                                                                                                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b> | <p>To further enhance and customize RTF documentation it is possible to create a custom master document, which can be used to add a table of contents, table of figures, headers and footers and to refresh linked files. In addition it is possible to create documents with sustainable links to generated 'pieces' of Enterprise Architect output, pre-divided by Enterprise Architect using <b>bookmarks</b>.</p> <p>As an alternative to the <i>Word</i> master document, internal to Enterprise Architect, see <i>Virtual Documents</i>.</p> <p>In addition to creating Word master documents, you can:</p> <ul style="list-style-type: none"> <li><b>Open a Report in Microsoft Word</b></li> <li><b>Change Linked Images to Embedded</b></li> <li><b>Apply Other Features of Word</b></li> </ul> | <p><a href="#">RTF Bookmarks</a><sup>[1796]</sup></p> <p><a href="#">Virtual Documents</a><sup>[1774]</sup></p> <p><a href="#">Open a Report in Microsoft Word</a><sup>[1796]</sup></p> <p><a href="#">Change Linked Images to Embedded</a><sup>[1796]</sup></p> <p><a href="#">Other Features of Word</a><sup>[1798]</sup></p> |

**Notes:**

- You can develop a report using the combined facilities of Word and Enterprise Architect with few problems, as long as you leave definition of the section styles to the final stages in Enterprise Architect just prior to report generation. Word truncates the section bookmarks, as it uses a smaller field length

for sections. In Word, you can review and edit reports generated by Enterprise Architect, but you cannot import them back into Enterprise Architect without damaging the section style definition

### 17.1.5.1 Open a Report in Microsoft Word

#### Topics:

| Topic | Detail                                                                                                                                                                                                                             | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Usage | To open an RTF file in MS Word, simply load Word and open the file as a normal document. Word converts the file. If Word is the default handler of RTF files, then double-click on the output file to load up and view the report. |          |

#### Notes:

- If you have Word configured to view RTF files, you can also click on the **View Output** button on the **Generate RTF Documentation** dialog

### 17.1.5.2 Change Linked Images to Embedded

One of the options available when generating RTF documentation is the ability to store image files in a separate directory to the RTF document. If at a later stage it becomes desirable to embed the images in the RTF documentation, this is especially important when the document is to be distributed. If the images are stored in a separate directory recipients of document see only the placeholder of images rather than the actual images.

If you import an RTF document into Word with the images not embedded into the document, you have the option of breaking the links to the images and saving the image in the document.

To break image links in Word, follow the steps below

#### How To:

| Step | Action                                                        | See Also |
|------|---------------------------------------------------------------|----------|
| 1.   | Open the required RTF file in Word.                           |          |
| 2.   | Select the <b>Edit   Links</b> menu option.                   |          |
| 3.   | Highlight all links in the <b>Links</b> list.                 |          |
| 4.   | Select the <b>Save Picture in Document</b> checkbox.          |          |
| 5.   | Click on the <b>Break Link</b> button.                        |          |
| 6.   | When prompted, click on the <b>Yes</b> button to break links. |          |

Word breaks the links and saves copies of the images inside the document. You can distribute this document without the image directory.

### 17.1.5.3 RTF Bookmarks

Bookmarks are markers that are automatically placed in your rich text document when you generate it. You can create a master document in Word and link to sections of an Enterprise Architect report based on bookmarks. For example, a Word document might have a section for a small part of your component model. Using bookmarks you can generate a full component model, and then link into just one section of the report.

This way you can maintain a complex Word document from parts of Enterprise Architect reports. If you link into Enterprise Architect reports, then you can regenerate the report and refresh Word links to update the master document without having manually changed anything. For more information on refreshing links, see the **Refresh Links** topic.

Bookmarks are GUID-based numbers that can be created for packages, diagrams and elements. A package bookmark applies from the beginning of a package to the end, and includes all child packages and elements underneath.

To bookmark a Section of Enterprise Architect for RTF Documentation, follow the steps below

**How To:**

| Step | Action                                                                                                                                      | See Also |
|------|---------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | In the Enterprise Architect <b>Project Browser</b> , right-click on the package to include in the documentation. The context menu displays. |          |
| 2.   | Select the <b>Documentation   Copy RTF Bookmark</b> menu option to paste the package into the clipboard as a bookmark for use in Word.      |          |

To Insert a Bookmarked Section of an Enterprise Architect RTF Document into Word, follow the steps below

**How To:**

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See Also                                             |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| 1.   | Open the Word document and position the cursor at the point at which to insert the file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                      |
| 2.   | Select the Word <b>Insert   File</b> menu option. The <b>Insert</b> dialog displays.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                      |
| 3.   | Locate and click on the file to insert, then click on the <b>Range</b> button.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                      |
| 4.   | In the <b>Range</b> cell type or paste the information from the clipboard.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                      |
| 5.   | Click on the <b>OK</b> button.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                      |
| 6.   | <p>Click on the drop-down arrow next to the <b>Insert</b> button. Select the <b>Insert as Link</b> option.</p> <p>The <b>Insert</b> option sets a permanent copy; the <b>Insert as Link</b> option creates a link that is updateable on altering the source document. For <b>Insert as Link</b> to operate you must first set <b>Refresh Links</b>.</p> <p>Every package is bookmarked in the RTF document according to the following rules:</p> <ul style="list-style-type: none"> <li>• All alphabetic and numeric characters remain the same</li> <li>• All other characters (including spaces) are converted to underscores</li> </ul> <p>For example UC01: Use Case Model becomes UC01__Use_Case_Model.</p> | <a href="#">Refresh Links</a><br><small>1801</small> |

**Notes:**

- You cannot use RTF Bookmarking in **Master Document** elements, which effectively replace RTF Bookmarking in Word.

RTF Bookmarking requires each bookmark to be unique. When you generate a report with a standard

RTF template (including in a single Model Document element), each bookmark is unique and there is a 1:1 association between the Elements-details being generated and the elements in the repository. As Master Documents are intended to contain multiple sub-documents, the association ceases to be 1:1. There is no simple method that enables the generated data to be uniquely identified directly in association with the original element

**Learn More:**

- [Refresh Links](#) <sup>[1801]</sup>
- [Create a Master Document](#) <sup>[1776]</sup>

### 17.1.5.4 Other Features of Word

Word offers a considerable number of document enhancement tools to complete your project documentation. Here are some of the things you can do with Word in Enterprise Architect generated RTF documentation:

| Topic                       | Link                                                          |
|-----------------------------|---------------------------------------------------------------|
| Add a Table of Contents     | <a href="#">Add a Table of Contents</a> <sup>[1798]</sup>     |
| Add a Table of Figures      | <a href="#">Add a Table of Figures</a> <sup>[1798]</sup>      |
| Add Headers and Footers     | <a href="#">Add Headers and Footers</a> <sup>[1799]</sup>     |
| Manipulating Tables in Word | <a href="#">Manipulating Tables in Word</a> <sup>[1799]</sup> |
| Refresh Linked Files        | <a href="#">Refresh Linked Files</a> <sup>[1801]</sup>        |

**Notes:**

- Enterprise Architect provides the basic content for your document - use Word to add the presentation and linkages

#### 17.1.5.4.1 Add Table of Contents

Among the features of MS word that can be incorporated into generated Enterprise Architect reports is the option to include a table of contents. A table of contents can be used to aid navigation of documentation and enhance the readability of Enterprise Architect RTF reports. This option provides hyperlinks to the diagrams included in the RTF documentation in the electronic version, and page numbers for both the printed and electronic documentation.

To include a Table of Contents in the RTF documentation, follow the steps below

**How To:**

| Step | Action                                                                                                        | See Also |
|------|---------------------------------------------------------------------------------------------------------------|----------|
| 1    | Open the Enterprise Architect RTF report to which to add a Table of Contents in MS Word                       |          |
| 2    | Select the <b>Insert   Reference   Index and Tables</b> menu option                                           |          |
| 3    | Click on the Table of Contents tab to set the options that are available for formatting the table of contents |          |

The format of the table of contents is dependant on the heading levels created when the RTF is generated. To set the heading style for details in Enterprise Architect RTF documentation, see the RTF **Document Options** topic.

**Learn More:**

- [Document Options](#)<sup>[1767]</sup>

#### 17.1.5.4.2 Add Table of Figures

Among the features of MS word that can be incorporated into generated Enterprise Architect reports is the option to include a table of figures. A table of figures can be used to aid the navigation of the documentation and enhance the readability of Enterprise Architect RTF reports. This option provide hyperlinks to the diagrams included in the RTF documentation in the electronic version and page numbers for both the printed and electronic documentation.

To include a Table of Figures in the RTF documentation, follow the steps below

**How To:**

| Step | Action                                                                                                              | See Also |
|------|---------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Open the Enterprise Architect RTF report to which to add a Table of Figures in MS Word.                             |          |
| 2.   | Select the <b>Insert   Reference   Index and Tables</b> menu option.                                                |          |
| 3.   | Click on the <b>Table of Figures</b> tab to set the options that are available for formatting the table of figures. |          |

#### 17.1.5.4.3 Add Headers and Footers

Among the features of MS word that can be used to enhance the appearance of Enterprise Architect RTF reports is the ability to add headers and footers to the documentation.

To include headers and footers in the RTF documentation follow the steps below

**How To:**

| Step | Action                                                                                    | See Also |
|------|-------------------------------------------------------------------------------------------|----------|
| 1.   | In MS Word, open the Enterprise Architect RTF report to which to add headers and footers. |          |
| 2.   | Select the <b>View   Header and Footer</b> menu option.                                   |          |

This enables you to enter information into the header section and the footer section of the RTF Documentation.

#### 17.1.5.4.4 Manipulate Tables in Word

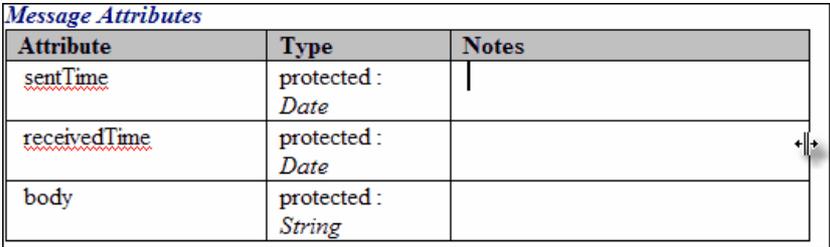
When generating RTF documentation from Enterprise Architect, tables are included when items such as **Attributes** and **Methods** are selected in the **For each Object** section in the **Rich Text Format Report** dialog. MS Word offers several levels of customization for tables and can be used to tidy the formatting of the tables in situations where the margins of the table exceed the dimensions of the page size selected in Word for printing.

### Resize Tables

When the amount of detail for a documented item such as an attribute or operation exceeds the margins of the page in MS Word it is necessary to manually resize the table in order to view all of the details.

To manually resize the table follow the steps below

#### How To:

| Step | Action                                                                                                                                                                                   | See Also |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Select the table that exceeds the margin size.                                                                                                                                           |          |
| 2.   | Mouse over the border of the table until the mouse pointer changes into the icon shown below.<br><br> |          |
| 3.   | Drag the cursor to the left to reduce the width of the table and then select the <b>File   Print Preview</b> menu option to confirm that the table borders are within the page margins.  |          |
| 4.   | Resize all of the tables that overhang the margins of the page by using the steps detailed above.                                                                                        |          |

### Applying Styles to Tables

One of the customizable properties of MS Word when working with tables is the ability to apply a style to a table, which enables you to rapidly change the appearance of the table.

To apply styles to a table, follow the steps below

#### How To:

| Step | Action                                                                                                | See Also |
|------|-------------------------------------------------------------------------------------------------------|----------|
| 1.   | Open the Enterprise Architect RTF report in which to change the table styles.                         |          |
| 2.   | Locate and select the table for which to adjust the appearance.                                       |          |
| 3.   | Select the <b>Table   Table Auto Format</b> menu option. The <b>Table Autoformat</b> dialog displays. |          |

From here you can specify a predefined table style from the **Table styles** list, or create a new style by clicking on the **New** button. The table styles defined in the **Table Autoformat** dialog only apply to one table at a time so you must apply the style to each table created individually.

#### 17.1.5.4.5 Refresh Links

If you link into Enterprise Architect reports, then you can regenerate the report and refresh MS Word links to update the Word master document without having manually changed anything.

To ensure that links are refreshed in the master document, you must select the **Update automatic links at Open** checkbox in Word.

To ensure that this setting is established follow the steps below

##### How To:

| Step | Action                                                                                       | See Also |
|------|----------------------------------------------------------------------------------------------|----------|
| 1.   | From within MS Word select the <b>Tools   Options</b> menu option.                           |          |
| 2.   | Select the <b>General</b> tab and select the <b>Update automatic links at Open</b> checkbox. |          |

## 17.2 HTML Reports



### Topics:

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also                                                |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <b>Abstract</b> | <p>Enterprise Architect provides automated web-based publishing of models. A new outline structure closely mirrors the model hierarchy and makes it very simple to explore models on-line. With a great new look and the ability to explore very large models efficiently on-line, the new web-publishing capability is a significant enhancement.</p> <p>Enterprise Architect enables export of an entire model or a single branch of the model to HTML Web pages. The <b>HTML report</b> provides an easy to use, highly detailed, Javascript based model tree. In addition, hyperlinked elements make browsing to related information very simple.</p> <p>The current implementation is based on internal and external templates and generated Javascript. The ability to edit all templates is to be added in a future version of Enterprise Architect.</p> | <a href="#">Create an HTML report</a> <sup>[1802]</sup> |

### Notes:

- You can create **Web Style Templates** to customize your HTML output
- You can use Master Documents and Model Documents to select specific areas of your model for reporting
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Documents** permission to generate HTML documents

### Learn More:

- [Web Style Templates](#)<sup>[1804]</sup>
- [Permission List](#)<sup>[205]</sup>

### 17.2.1 Create an HTML Report

To create an HTML report, follow the steps below

#### How To:

| Step | Action                                                                                                                                                           | See Also |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | <p>In the Project Browser, right-click on the root package for the report (all child packages are included in the output).</p> <p>The context menu displays.</p> |          |

| Step | Action                                                                                                                    | See Also                                                      |
|------|---------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| 2    | Select the <b>Documentation   HTML Report</b> menu option.<br>The Generate HTML Report dialog displays.                   | <a href="#">Generate HTML Report dialog</a> <sup>[1804]</sup> |
| 3    | In the <b>Output to</b> field, select an output directory for your report.<br>Set any other required options.             |                                                               |
| 4    | Click on the <b>Generate</b> button to generate the report.<br>The <b>Progress</b> field shows total percentage complete. |                                                               |
| 5    | Once the report is complete, click on the <b>View</b> button to launch your default HTML viewer and view the web pages.   |                                                               |

**Topics:**

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also                                                                                                                                       |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Compatibility</b> | <p>The web report produced is compatible with any standard web server, on either Unix or Windows platforms. Simply bundle up the entire output directory and place it within the context of your web server. All path names should be relative and case sensitive.</p> <p>When you view the HTML report in your HTML viewer, you can switch directly to a page for a specific diagram or element by specifying the appropriate GUID after the report web address. That is:</p> <pre>ht t p : // pat h/ pat h/ pat h/ I ndex . ht m? gui d=xxxxxxxxxxxxxx</pre> <p>The word <i>guid</i> must be in lower case, and the value must not include braces { }. For example:</p> <pre>ht t p : // . . . / pat h/ I ndex . ht m?gui d=DC62B0DA- 0D60-4447-85E6-B9BBAE7FC90F</pre> <p>You can obtain the GUID for the diagram or element using the <b>Copy Reference</b> option on the Project Browser <b>Diagram</b>, <b>Element</b> and <b>Package</b> context menu options.</p> <p>The page locator does not work directly in Internet Explorer. Firefox automatically converts the path to <b>file:///C:/path</b> protocol, and actions it. That protocol also works in Internet Explorer. Therefore, to use the absolute references without a web server you must access the path using the <b>file:///</b> protocol.</p> | <a href="#">Diagram Menu</a> <sup>[45]</sup><br><a href="#">Element Menu</a> <sup>[448]</sup><br><a href="#">Package Menu</a> <sup>[444]</sup> |
| <b>Quick Start</b>   | To generate an HTML report right now, follow the steps above on the System Model package of the EAExample project.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                |

**Notes:**

- If you are using Microsoft Internet Explorer 7.0 or later, and you do not have it open, its security profile might block the report display; click on the explanation banner at the top of the screen and select the **Allow Blocked Content** context menu option

### 17.2.2 The Generate HTML Report Dialog

The **Generate HTML Report** dialog is used to generate documentation about your model in HTML format. There are various settings to choose from to control the output, as described below.

| Field                                  | Usage                                                                                                                                                                              | See also                                              |
|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| <b>Package</b>                         | Display the name of the package you are creating documentation for.                                                                                                                |                                                       |
| <b>Title</b>                           | Type the title for your HTML documentation; defaults to <b>Package</b> .                                                                                                           |                                                       |
| <b>Output to</b>                       | Type the directory path your documentation is saved to.                                                                                                                            |                                                       |
| <b>Style</b>                           | Select a web style template to apply to your documentation (optional).                                                                                                             | <a href="#">Web Style Templates</a> <sup>[1804]</sup> |
| <b>File</b>                            | Specify the file extension for your HTML documentation files; the default is <b>.htm</b> .                                                                                         |                                                       |
| <b>Header Image</b>                    | Enter or select the file path for the header image graphic to display on your HTML output. If you do not specify a file path, the image defaults to the Enterprise Architect logo. |                                                       |
| <b>Preserve White space in Notes</b>   | Preserve existing white space in your notes; deselect to remove white space.                                                                                                       |                                                       |
| <b>No page for Note and Text items</b> | Omit the page for your notes and text items in the HTML report.                                                                                                                    |                                                       |
| <b>Default Diagram</b>                 | Select the diagram the report should open to when the generated documentation is loaded.                                                                                           |                                                       |
| <b>Image Format</b>                    | Select the image file format to save your images in, either PNG or GIF.                                                                                                            |                                                       |
| <b>Include</b>                         | Select each area of your model to include in your report.                                                                                                                          |                                                       |
| <b>System</b>                          | Select each section to generate in your report.                                                                                                                                    |                                                       |

Click on the **Generate** button to generate the HTML report with the settings you have defined.

Click on the **View** button to display the report you have generated.

#### Notes:

- You can also select packages in your model to exclude them (and their subordinate packages) from the HTML reports you generate

#### Learn more:

- [Exclude Package from Report](#) <sup>[1731]</sup>

### 17.2.3 Web Style Templates

The *HTML and CSS Style Editor* enables you to edit the HTML associated with various sections of the HTML Report facility in Enterprise Architect. You would typically use this functionality to customize a report's look and feel for your company or client. The editor is derived from, and provides the facilities of, the common Code Editor.

**How To:**

To create or edit web style templates, follow the steps below:

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Open the Resources window ( <b>View   More Project Tools   Project Resources</b> ) and expand the <i>Templates</i> folder.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |
| 2    | <p>To:</p> <ul style="list-style-type: none"> <li>• Create a new template, right-click on the Web Style Templates folder and select the <b>Create HTML Template</b> context menu option; enter a name for the new template when prompted to do so.</li> <li>• Edit an existing template, expand the Web Style Templates folder and either double-click on the template name or right-click and select the <b>Modify HTML Style Template</b> context menu option</li> </ul> <p>In either case, the <b>HTML and CSS Style Editor</b> displays.</p> <p>To delete a template, right-click on it and select the Delete HTML Template context menu option.</p> |          |

**Topics:**

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>Usage</b> | <p>The <b>HTML and CSS Style Editor</b> contains a list of all available HTML fragments for modification and customization.</p> <p>Each fragment typically contains HTML plus one or more special tag names that Enterprise Architect replaces with information during generation. Currently you cannot alter the content within the tag names, but you can omit a complete tag by removing it, or alter its basic display properties in the surrounding HTML.</p> <p>Special tag names are delimited by # characters - for example, <b>#NOTES#</b>.</p> <p>The:</p> <ul style="list-style-type: none"> <li>• <b>Get Default</b> button retrieves the default Enterprise Architect template for the currently selected template item in the left hand list</li> <li>• <b>Save</b> button saves your version of the template for this style only</li> <li>• <b>Delete</b> button removes your modified version of the template, which causes Enterprise Architect to use the default template during report generation</li> </ul> <p>To select a template during generation, use the Style drop-down arrow on the <b>Generate HTML Report dialog</b>. Once a style is selected, Enterprise Architect applies that to the current report. Select &lt;default&gt; for the inbuilt style.</p> | <a href="#">HTML Template Fragments</a><br><small>[1805]</small> |

**Notes:**

- Each time Enterprise Architect generates the web report it overwrites these files, so you must back up your modified versions and copy them back in after every update

**Learn More:**

- [Code Editor](#)[1394]

## 17.2.4 HTML Template Fragments

This topic identifies each of the HTML Template fragments available through the **HTML and CSS Style Editor**, and defines the fields of each fragment.

| Section               | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Body - Diagram</b> | <ul style="list-style-type: none"> <li>• #AUTHOR# - Diagram author</li> <li>• #CREATEDATE# - Diagram created date</li> <li>• #CSS# - Style Sheet to use</li> <li>• #GUID# - Diagram GUID</li> <li>• #IMAGE# - Image of the diagram</li> <li>• #IMAGES# - Image path</li> <li>• #LOCKED# - Is the diagram locked (<b>true</b> or <b>false</b>)</li> <li>• #MODDATE# - Diagram modified date</li> <li>• #NAME# - Diagram name</li> <li>• #NOTE# - Diagram notes</li> <li>• #STEREOTYPE# - Diagram stereotype</li> <li>• #TITLE# - Diagram title</li> <li>• #TYPE# - Diagram type</li> <li>• #VERSION# - Diagram version</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Body – Object</b>  | <ul style="list-style-type: none"> <li>• #ABSTRACT# - Element abstract (if true, = <b>abstract</b>)</li> <li>• #ACTIVE# - Element isActive (<b>true</b> or <b>false</b>)</li> <li>• #ALIAS# - Element alias ("<b>#ALIAS#</b>")</li> <li>• #AUTHOR# - Element author</li> <li>• #BEHAVIOR# - Returns the object behavior.</li> <li>• #CLASSIFIER# - Element classifier</li> <li>• #CLASSIFIERREF# - Returns the HREF of the classifier</li> <li>• #COMPLEXITY# - Element complexity</li> <li>• #CREATEDATE# - Element created date</li> <li>• #CSS# - Style Sheet to use</li> <li>• #DIAGRAMS# - List of diagrams the element is on</li> <li>• #DIFF# - Element difficulty</li> <li>• #GUID# - Element GUID</li> <li>• #IMAGES# - Image path</li> <li>• #KEYWORDS# - Element keywords</li> <li>• #LANGUAGE# - Element language</li> <li>• #LEAF# - Element isLeaf (<b>true</b> or <b>false</b>)</li> <li>• #LEVELNUMBER# - Element level number</li> <li>• #MODDATE# - Element modified date</li> <li>• #MULTIPLICITY# - Element multiplicity ("<b>Multiplicity: #MULTIPLICITY#</b>")</li> <li>• #NAME# - Element name</li> <li>• #NOTE# - Element notes</li> <li>• #PHASE# - Element phase</li> <li>• #PRIORITY# - Element priority</li> <li>• #ROOT# - Element isRoot (<b>true</b> or <b>false</b>)</li> <li>• #SCOPE# - Element scope</li> <li>• #STATUS# - Element status</li> <li>• #STEREOTYPE# - Element stereotype(s)</li> <li>• #TYPE# - Element type; for example, Class, Object</li> <li>• #VERSION# - Element version</li> </ul> |

| Section                      | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Body – Project Glossary      | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Glossary Item</b> for each Project Glossary item</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Body - Project Glossary Item | <ul style="list-style-type: none"> <li>• #MEANING# - Glossary Meaning</li> <li>• #TERM# - Glossary Term</li> <li>• #TYPE# - Glossary Type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Body – Project Issue         | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Issue Item</b> for each Project Issue item</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Body - Project Issue Item    | <ul style="list-style-type: none"> <li>• #DATERESOLVED# - Project Issue resolved date (blank if no date entered)</li> <li>• #ISSUE# - Project Issue name</li> <li>• #ISSUEDATE# - Project Issue issue date</li> <li>• #NOTES# - Project Issue notes</li> <li>• #OWNER# - Project Issue owner</li> <li>• #RESOLUTION# - Project Issue resolution</li> <li>• #RESOLVER# - Project Issue resolver</li> <li>• #STATUS# - Project Issue status</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Body – Project Task          | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Task Item</b> for each Project Task item</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Body - Project Task Item     | <ul style="list-style-type: none"> <li>• #ENDDATE# - Project Task end date</li> <li>• #NAME# - Project Task name</li> <li>• #NOTES# - Project Task notes</li> <li>• #OWNER# - Project Task owner</li> <li>• #PHASE# - Project Task phase</li> <li>• #PRIORITY# - Project Task priority</li> <li>• #STARTDATE# - Project Task start date</li> <li>• #STATUS# - Project Task status</li> <li>• #TYPE# - Project Task type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Content – Attributes         | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Attributes Item</b> for each attribute on an element</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Content – Attributes Item    | <ul style="list-style-type: none"> <li>• #ALIAS# - Attribute alias ("<i>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</i>")</li> <li>• #CONST# - Attribute is constant value ("<b>const</b> " &lt;-- Note Space)</li> <li>• #CONSTRAINT# - Attribute constraint</li> <li>• #DEFAULT# - Attribute default value ("<i>&lt;i&gt;Initial:&lt;/i&gt; #DEFAULT&lt;br&gt;</i>")</li> <li>• #NAME# - Attribute name</li> <li>• #NOTE# - Attribute notes</li> <li>• #ORDERED# - Attribute is ordered value ("<b>Ordered &lt;br /&gt;</b>")</li> <li>• #RANGE# - If lower != 1 ("<b>Range:&lt;lower&gt; to &lt;upper&gt;</b>")</li> <li>• #SCOPE# - Attribute scope ("<b>#SCOPE#</b> " &lt;-- Note space)</li> <li>• #STATIC# - Attribute is static value ("<b>static</b> " &lt;-- Note Space)</li> <li>• #STEREOTYPE# - Attribute stereotype(s)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>")</li> <li>• #TYPE# - Attribute type (if <b>Column</b>, "<b>#TYPE(Column Precision,</b></li> </ul> |

| Section                             | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                     | Scale)")                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Content – Constraints               | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Constraints Item</b> for each constraint on an element</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Content – Constraints Item          | <ul style="list-style-type: none"> <li>• #CONSTRAINT# - Constraint name</li> <li>• #NOTES# - Constraint notes</li> <li>• #STATUS# - Constraint status</li> <li>• #TYPE# - Constraint type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Content – Custom Properties         | <ul style="list-style-type: none"> <li>• #CONTENT# - List of all <b>Content – Custom Properties - Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Content – Custom Properties - Item  | <ul style="list-style-type: none"> <li>• #PROPERTY# - Custom property name</li> <li>• #VALUE# - Custom property value<br/>( if type equal boolean<br/>true or false<br/>else<br/>#VALUE#)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Content – Files                     | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Files Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Content – Files Item                | <ul style="list-style-type: none"> <li>• #DATE# - File date</li> <li>• #NAME# - File filename (<b>&lt;a href="#"#FILENAME#" &lt;/a&gt;</b>)</li> <li>• #NOTES - File notes</li> <li>• #SIZE# - File size</li> <li>• #TYPE# - File type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Content - Inherited Attributes      | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Inherited Attributes Item</b> for each attribute on an element</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Content - Inherited Attributes Item | <ul style="list-style-type: none"> <li>• #ALIAS# - Attribute alias ("<b>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</b>")</li> <li>• #CONST# - Attribute is constant value ("<b>const</b> " &lt;-- Note Space)</li> <li>• #CONSTRAINT# - Attribute constraint</li> <li>• #DEFAULT# - Attribute default value ("<b>&lt;i&gt;Initial:&lt;/i&gt; #DEFAULT&lt;br&gt;</b>")</li> <li>• #NAME# - Attribute name</li> <li>• #NOTE# - Attribute notes</li> <li>• #ORDERED# - Attribute is ordered value ("<b>Ordered &lt;br /&gt;</b>")</li> <li>• #RANGE# - If lower != 1 ("<b>Range:&lt;lower&gt; to &lt;upper&gt;</b>")</li> <li>• #SCOPE# - Attribute scope ("<b>#SCOPE#</b> " &lt;-- Note space)</li> <li>• #STATIC# - Attribute is static value ("<b>static</b> " &lt;-- Note Space)</li> <li>• #STEREOTYPE# - Attribute stereotype(s)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>" )</li> <li>• #TYPE# - Attribute type (if <b>Column</b>, "<b>#TYPE(Column Precision, Scale)</b>")</li> </ul> |
| Content - Inherited Operations      | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Inherited Operations Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Content - Inherited Operations Item | <ul style="list-style-type: none"> <li>• #ABSTRACT# - Operation abstract (<b>abstract</b>)</li> <li>• #ALIAS# - Operation alias ("<b>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</b>")</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Section                              | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                      | <ul style="list-style-type: none"> <li>• #CONCURRENCY# - Operation concurrency (<b>blank</b> if not set)</li> <li>• #CONST# - Operation constant (<b>const</b>)</li> <li>• #CONSTRAINTS# - List of <b>Method Constraint</b></li> <li>• #ISQUERY# - Operation IsQuery (<b>isQuery</b>)</li> <li>• #NAME# - Operation name</li> <li>• #NOTE# - Operation notes.</li> <li>• #PARAMS# - List of <b>Content – Operations Item Parameters</b></li> <li>• #SCOPE# - Operation Scope</li> <li>• #STATIC# - Operation IsStatic (<b>static</b>)</li> <li>• #STEREOTYPE# - Operation stereotype</li> <li>• #TAGLABEL# - static text (<b>Tags</b>)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>")</li> <li>• #TYPE# - Operation type</li> </ul>                                                                                                                                                          |
| Content – Notes                      | <ul style="list-style-type: none"> <li>• #VALUE#&gt; - Notes text</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Content – Operations                 | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Operations Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Content – Operations Item            | <ul style="list-style-type: none"> <li>• #ABSTRACT# - Operation abstract (<b>abstract</b>)</li> <li>• #ALIAS# - Operation alias ("<b>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</b>")</li> <li>• #CONCURRENCY# - Operation concurrency (<b>blank</b> if not set)</li> <li>• #CONST# - Operation constant (<b>const</b>)</li> <li>• #CONSTRAINTS# - List of <b>Method Constraint</b></li> <li>• #ISQUERY# - Operation IsQuery (<b>isQuery</b>)</li> <li>• #NAME# - Operation name</li> <li>• #NOTE# - Operation notes.</li> <li>• #PARAMS# - List of <b>Content – Operations Item Parameters</b></li> <li>• #SCOPE# - Operation Scope</li> <li>• #STATIC# - Operation IsStatic (<b>static</b>)</li> <li>• #STEREOTYPE# - Operation stereotype</li> <li>• #TAGLABEL# - static text (<b>Tags</b>)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>")</li> <li>• #TYPE# - Operation type</li> </ul> |
| Content – Operations Item Parameters | <ul style="list-style-type: none"> <li>• #DEFAULT# - Op Parameter default</li> <li>• #GUID# - Op Parameter GUID</li> <li>• #KIND# - Op Parameter kind</li> <li>• #NAME# - Op Parameter name</li> <li>• #NOTES# - Op Parameter notes</li> <li>• #STEREOTYPE# - Op Parameter stereotype</li> <li>• #TYPE# - Op Parameter type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Content – Resource Allocation        | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content – Resource Allocation Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Content – Resource Allocation Item   | <ul style="list-style-type: none"> <li>• #ACTUAL# - Resource actual time</li> <li>• #ENDDATE# - Resource end date</li> <li>• #EXPECTED# - Resource expected date</li> <li>• #NOTES# - Resource notes</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Section                 | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | <ul style="list-style-type: none"> <li>• #PERCENT# - Resource percent complete</li> <li>• #RESOURCE# - Resource name</li> <li>• #ROLE# - Resource role</li> <li>• #STARTDATE# - Resource start date</li> <li>• #TIME# - Resource time</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                |
| CSS – Main              | <ul style="list-style-type: none"> <li>• None</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| CSS – ToC               | <ul style="list-style-type: none"> <li>• None</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Feature Notes           | <ul style="list-style-type: none"> <li>• #FIELD#</li> <li>• #VALUE#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Javascript              | <ul style="list-style-type: none"> <li>• None</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Link (Association)      | <ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Link (Association) Item | <ul style="list-style-type: none"> <li>• #CONNECTION# - Connector type</li> <li>• #IMAGE# - The file path of the images</li> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SOURCEROLE# - Source role</li> <li>• #SOURCEROLENOTES# - Source role notes</li> <li>• #STEREOTYPE# - Connector stereotype</li> <li>• #TARGETROLE# - Target role</li> <li>• #TARGETROLENOTES# Target role notes</li> <li>• #TYPE# - Connector type</li> </ul>                                              |
| Link (Flow)             | <ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Link (Flow) Item        | <ul style="list-style-type: none"> <li>• #DIRECTION# - Connector direction</li> <li>• #ELEMNAME# - Name of the element at the source/destination of the connector</li> <li>• #IMAGE# - The file path of the images</li> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #LINKREF# - Page name of the element at the source/destination of the connector</li> <li>• #NAME# - Connector name</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #STEREOTYPE# - Connector stereotype</li> <li>• #TYPE# - Connector type</li> </ul> |
| Link (Other)            | <ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Link (Other) Item       | <ul style="list-style-type: none"> <li>• #CONNECTION# - Connector type</li> <li>• #IMAGE# - The file path of the images</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Section                 | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | <ul style="list-style-type: none"> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SOURCEROLE# - Source role</li> <li>• #SOURCEROLENOTES# - Source role notes</li> <li>• #STEREOTYPE# - Connector stereotype</li> <li>• #TARGETROLE# - Target role</li> <li>• #TARGETROLENOTES# Target role notes</li> <li>• #TYPE# - Connector type</li> </ul>                                                                                                                                                                                   |
| Linked Document         | <ul style="list-style-type: none"> <li>• #LINKDOC# - Linked Document.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Linked Requirement      | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Linked Requirement Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Linked Requirement Item | <ul style="list-style-type: none"> <li>• #DIFF# - Linked Requirement difficulty</li> <li>• #NAME# - Linked Requirement name</li> <li>• #PRIORITY# - Linked Requirement priority</li> <li>• #STATUS# - Linked Requirement status</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Linked Section          | <ul style="list-style-type: none"> <li>• #ITEMS#</li> <li>• #TITLE#</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Maintenance             | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Maintenance Line Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Maintenance Line Item   | <ul style="list-style-type: none"> <li>• #DATEREPORTED# - Maintenance date reported</li> <li>• #DATERESOLVED# - Maintenance date resolved</li> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Maintenance notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #PRIORITY# - Maintenance priority</li> <li>• #PROBLEM# - Maintenance name</li> <li>• #REPORTEDBY# - Maintenance reported by</li> <li>• #RESOLVEDBY# - Maintenance resolved by</li> <li>• #RESOLVERNOTES# - Maintenance resolved notes</li> <li>• #STATUS# - Maintenance status</li> <li>• #TYPE# - Maintenance type</li> <li>• #VERSION# - Maintenance version</li> </ul> |
| Message                 | <p>(Applies only to Sequence messages.)</p> <ul style="list-style-type: none"> <li>• #CONTENT# - Loops the <b>Message Item</b> for each attribute on an element</li> <li>• #DIRECTION# - Contains the value <b>To</b> or <b>From</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Message Item            | <ul style="list-style-type: none"> <li>• #KIND# - The Message <b>Kind</b> field</li> <li>• #MESSAGE# - Connector Message</li> <li>• #NAME# - Name of the Message (&lt;a href="&lt;path&gt;"&gt;#NAME#) If Message has a classifier: #NAME#=#NAME# :Classifier"</li> <li>• #NOTES# - The Message notes (&lt;strong&gt;Type:&lt;/strong&gt; #Item Type#&lt;br /&gt; #NOTES#)</li> </ul>                                                                                                                                                                                                                                                                                                      |

| Section                 | Fields                                                                                                                                                                                                                                                                                                                                    |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | <ul style="list-style-type: none"> <li>• #SYNCH# - The Message <b>Synch</b> field</li> <li>• #TYPE# - The type of Message</li> </ul>                                                                                                                                                                                                      |
| Method Constraint       | <ul style="list-style-type: none"> <li>• #NAME# - Method Constraint name</li> <li>• #NOTES# - Method Constraint notes</li> <li>• #TYPE# - Method Constraint type</li> </ul>                                                                                                                                                               |
| Object Requirement      | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Object Requirement Item</b></li> </ul>                                                                                                                                                                                                                                    |
| Object Requirement Item | <ul style="list-style-type: none"> <li>• #DIFF# - Requirement difficulty</li> <li>• #NAME# - Requirement name</li> <li>• #NOTES# - Requirement notes</li> <li>• #PRIORITY# - Requirement priority</li> <li>• #STABILITY# - Requirement stability</li> <li>• #STATUS# - Requirement status</li> <li>• #TYPE# - Requirement type</li> </ul> |
| Package Content         | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Package Content Row</b></li> </ul>                                                                                                                                                                                                                                        |
| Package Content Row     | <ul style="list-style-type: none"> <li>• #NAME# - Link to Package (&lt;a href="#Link to file#"&gt;#Package name#&lt;/a&gt;)</li> <li>• #TYPE# - Link to Image (&lt;img src="#path to image#"&gt;)</li> </ul>                                                                                                                              |
| Page - Basic template   | <ul style="list-style-type: none"> <li>• #CONTENT# - Contains <b>Body - Diagram</b> through to <b>Body - Object</b></li> <li>• #TITLE# - Current package name</li> </ul>                                                                                                                                                                  |
| Page - Index            | <ul style="list-style-type: none"> <li>• #CSS# - Style Sheet to use</li> <li>• #HOME# - A link to the Start page</li> <li>• #JS# - Javascript to use</li> <li>• #TITLE# - Current package name</li> <li>• #TOC# - <b>To be established</b></li> </ul>                                                                                     |
| Page - ToC              | <ul style="list-style-type: none"> <li>• None</li> </ul>                                                                                                                                                                                                                                                                                  |
| Scenario                | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Scenario Item</b></li> <li>• #EXCEPTIONS# - List of Structured Scenario exceptions</li> <li>• #STRUCTURED# - List of Structured Scenarios</li> </ul>                                                                                                                      |
| Scenario Item           | <ul style="list-style-type: none"> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Scenario notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SCENARIO# - Scenario name</li> <li>• #TYPE# - Scenario type</li> </ul>                                                                 |
| Scenario Exception      | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Scenario Exception Item</b> for each exception</li> </ul>                                                                                                                                                                                                                   |
| Scenario Exception Item | <ul style="list-style-type: none"> <li>• #NAME# - Exception name</li> <li>• #STEPNO# - Exception step number</li> </ul>                                                                                                                                                                                                                   |

| Section                   | Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           | <ul style="list-style-type: none"> <li>• #TYPE# - Exception Type</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Scenario Structured       | <ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Scenario Structured Items</b> for each Structured Scenario item</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Scenario Structured Items | <ul style="list-style-type: none"> <li>• #ACTION# - Name of the scenario</li> <li>• #STEPNO# - Scenario step number</li> <li>• #RESULT# - Step result value</li> <li>• #USES# - Step uses value</li> <li>• #STATE# - Step state value</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Tagged Value              | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Tagged Value Line Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Tagged Value Line Item    | <ul style="list-style-type: none"> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Tagged Value notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #PROPERTY# - Tagged Value name</li> <li>• #VALUE# - Tagged Value if type is boolean (value is <b>true</b> or <b>false</b>)</li> </ul>                                                                                                                                                                                                                                                                                                                                               |
| Test Cases                | <ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Test Cases Line Item</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Test Cases Line Item      | <ul style="list-style-type: none"> <li>• #ACCEPTANCE# - Test case acceptance notes</li> <li>• #CHECKEDBY# - Test case checked by</li> <li>• #CLASS# - Test case class (<b>Unit, Integration, System, Acceptance, Scenario</b>)</li> <li>• #IMAGE# - The file path of the images</li> <li>• #INPUT# - Test case input notes</li> <li>• #NOTES# - Test case notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #RESULTS# - Test case result notes</li> <li>• #RUNBY# - Test case run by</li> <li>• #RUNDATE# - Test case last run</li> <li>• #STATUS# - Test case status</li> <li>• #TEST# - Test case name</li> <li>• #TYPE# - Test case type</li> </ul> |

Learn More:

- [HTML and CSS Style Editor](#)<sup>[1804]</sup>

## 17.2.5 Master Documents and Model Documents

Enterprise Architect enables you to generate HTML documentation using Master Document and Model Document elements, just as you do for **RTF reports**. However, there are two differences:

- You do not specify the template in the document elements; you must select it on the Generate HTML Report dialog, in the **Style** field
- You do not use the *SearchName* and *SearchValue* Tagged Values in the Model Document elements to select the generated elements

**Learn More:**

- [Virtual Documents](#)<sup>[1774]</sup>
- [Generate HTML Report Dialog](#)<sup>[1804]</sup>

**17.2.5.1 Generate a Master Document (HTML)**

To generate the HTML documentation defined in the Master Document and/or Model Documents, follow the steps below

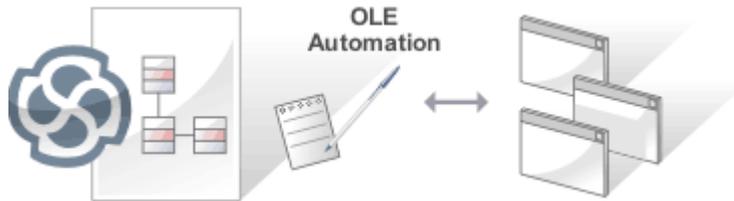
**How To:**

| Step | Action                                                                                                                                                                                                                                                                                                                             | See Also                                                      |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| 1    | On the documentation diagram, click on the Master Document element (or on an independent Model Document element).                                                                                                                                                                                                                  |                                                               |
| 2    | Select the <b>Project   Documentation   HTML Report</b> option.<br>The Generate HTML Report dialog displays.                                                                                                                                                                                                                       |                                                               |
| 3    | Set the options for your HTML document as required. Ensure that you select your Master Document template in the <b>Style</b> field.                                                                                                                                                                                                | <a href="#">Generate HTML Report Dialog</a> <sup>[1804]</sup> |
| 4    | Click on the <b>Generate</b> button to create the documentation.<br>The HTML Report Generator works through the defined content of the Master Document element and/or the Model Document elements and pulls in the information from the listed packages, formatted according to the template identified in the <b>Style</b> field. |                                                               |
| 5    | Click on the <b>View</b> button to view the documentation.                                                                                                                                                                                                                                                                         |                                                               |

**Part**



## 18 Automation



This section describes how you can automate and extend the facilities of Enterprise Architect through:

| Topic                                 | Link                                                                    |
|---------------------------------------|-------------------------------------------------------------------------|
| Scripts                               | <a href="#">Scripts</a> <sup>[1817]</sup>                               |
| The Enterprise Architect Object Model | <a href="#">The Enterprise Architect Object Model</a> <sup>[1822]</sup> |
| The Enterprise Architect Add-In Model | <a href="#">The Enterprise Architect Add-In Model</a> <sup>[1967]</sup> |

## 18.1 Scripting



### Topics:

| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                                                                                                                                                                                                         |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b> | <p>Scripts executed in Enterprise Architect have access to the currently open model and are a powerful tool for querying and updating the model in situations that would otherwise require you to perform time consuming and repetitive GUI tasks. Enterprise Architect supports management of scripts using the following script engines:</p> <ul style="list-style-type: none"> <li>• JavaScript</li> <li>• Microsoft JScript</li> <li>• Microsoft VBScript</li> </ul> <p>The management interface for Scripting is the <b>Scripting</b> window, which contains the:</p> <ul style="list-style-type: none"> <li>• <b>Script Tree View (Scripts tab)</b>, which you use to review, create and edit scripts</li> <li>• <b>Script Console (Console tab)</b>, which you use to operate on an executing script.</li> </ul> <p>Scripts are managed in groups. The first group in the list is always Local Scripts, which are available to any model. You cannot create, edit, drag-and-drop or delete these scripts. Local Scripts are files in the Scripts subdirectory of the Enterprise Architect installation - any instance of Enterprise Architect that has a currently open model can see these scripts.</p> <p>All other groups are User Scripts, which you create yourself. A user group can be <b>one of five types</b>, each of which applies a template and certain conditions to the scripts you create within that group. User scripts are only visible inside the model in which they were created; the contents of the scripts are stored with the model, although they can be saved to the file system easily using the <b>Script Editor</b>.</p> | <p><a href="#">Script Tab</a> <sup>[1818]</sup></p> <p><a href="#">Console Tab</a> <sup>[1820]</sup></p> <p><a href="#">Script Group Properties</a> <sup>[1819]</sup></p> <p><a href="#">Script Editor</a> <sup>[1407]</sup></p> |

### Notes:

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- If you intend to use the Scripting facility under Crossover/WINE, you must also install Internet Explorer version 6.0 or above

### 18.1.1 Scripts Tab

#### Topics:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also                                                                                                            |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b>         | <p>The <b>Scripts</b> tab is composed of a toolbar and a view of all scripts according to group.</p> <p>To execute a script, press ( <b>Ctrl</b> ) while you double-click on the script name.</p> <p>To write or edit a user script, double-click on the script name to display the <b>Script Editor</b>. This usually displays a script template, determined by the user group type as assigned on creation (see below) or on the <b>Script Group Properties</b> dialog.</p> <p>You can move or copy a script from one user scripts folder to another. To:</p> <ul style="list-style-type: none"> <li>• Move a script, highlight it in the <b>Scripts</b> tab and drag it into the user scripts folder it now belongs to</li> <li>• Copy a script, highlight it in the <b>Scripts</b> tab and press ( <b>Ctrl</b> ) while you drag it into the user scripts folder in which to duplicate it</li> </ul> | <a href="#">Script Editor</a> <sup>[1407]</sup><br><a href="#">Script Group Properties dialog</a> <sup>[1819]</sup> |
| <b>Context Menus</b> | <p>The script groups and their scripts also have context menus that provide some or all of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Group Properties</b> - to display or edit script group properties in the <b>Script Group Properties</b> dialog</li> <li>• <b>Run Script</b> - to execute the selected script</li> <li>• <b>Rename Script</b> - to change the name of the selected group or script</li> <li>• <b>New VBScript/JScript/JavaScript</b> - add a new script to the selected user group</li> <li>• <b>Import Workflow Script</b> - to display the Browser dialog through which you locate and select a workflow script source (.vbs) file to import into the Workflow script folder</li> <li>• <b>Delete Group/Script</b> - to delete the selected user group or script</li> </ul>                                                                          | <a href="#">Script Group Properties</a> <sup>[1819]</sup>                                                           |

The **Scripts** tab toolbar provides the following options:

#### Reference:

| Icon                                                                                | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See also |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|  | <p>Create a new script group. This option displays a short menu of the types of script group you can create, namely:</p> <ul style="list-style-type: none"> <li>• Normal group (  )</li> <li>• Project Browser group (  )</li> <li>• Diagram group (  )</li> <li>• Workflow Group (  )</li> <li>• Search Group (  )</li> </ul> <p>The new group is added to the end of the list in the <b>Scripting</b> window, with the '</p> |          |

| Icon                                                                                | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See also                                                                                         |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
|                                                                                     | <i>New group</i> ' text highlighted so that you can type in the group name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                  |
|    | <p>Create a new script file in the selected script group; displays a short menu of the types of script you can create, namely:</p> <ul style="list-style-type: none"> <li>• VBScript (  )</li> <li>• JScript (  )</li> <li>• JavaScript (  )</li> </ul> <p>The new script is added to the end of the list in the selected group, with the '<i>New script</i>' text highlighted so that you can type in the script name.</p> |                                                                                                  |
|    | Refresh the script tree in the <b>Scripting</b> window. This icon also reloads any changes made to a workflow script.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                  |
|    | Compile and execute the selected script. The output from the script is written to the <b>Script</b> tab of the <b>Output</b> window, which you display using the <b>View Script Output</b> button (below).                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                  |
|    | Stop an executing script. The icon is disabled if no script is executing.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                  |
|    | <p>Delete a <i>script</i> from the model. You cannot use this icon to delete a script <i>group</i> (see the Context Menu section, below), scripts in the Local Scripts group, or a script that is executing.</p> <p>The system prompts you to confirm the deletion only if the <b>Confirm Deletes</b> checkbox is selected in the <b>Project Browser</b> panel of the General page of the <b>Options</b> dialog. If this option is not selected, no prompt is displayed. Script deletion is permanent - scripts cannot be recovered.</p>                                                                                                                                       | <a href="#">Script Tab</a> <sup>[1818]</sup><br><a href="#">General setting</a> <sup>[422]</sup> |
|  | Display the <b>Output</b> window with the results of the most recently executed script displayed in the <b>Script</b> tab.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                  |

**Notes:**

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- If you select to delete a script group that contains scripts, the system always prompts you to confirm the action regardless of any system settings for delete operations. Be certain that you intend to delete the group and its scripts before confirming the deletion - deletion of script groups and scripts is permanent

**18.1.1.1 Script Group Properties****Topics:**

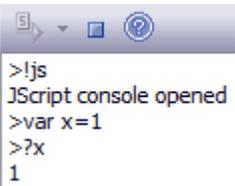
| Topic        | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                                                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b> | <p>The <b>Script Group Properties</b> dialog enables you to set the following properties of the script group:</p> <ul style="list-style-type: none"> <li>• <b>Name</b> - The name of the script group; the dialog also displays the automatically-assigned GUID of the group</li> <li>• <b>Group Type</b> - The type of scripts contained in the group. This can be one of: <ul style="list-style-type: none"> <li>• <b>Normal</b> - (  ) Ordinary scripts</li> <li>• <b>Project Browser</b> - (  ) Scripts that are listed in and can</li> </ul> </li> </ul> | <a href="#">Package Context Menu</a> <sup>[444]</sup><br><a href="#">Workflow Scripts</a> <sup>[231]</sup><br><a href="#">Model Search</a> <sup>[475]</sup><br><a href="#">Diagram Context Menu</a> <sup>[538]</sup> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>be executed from the Project Browser <b>Scripts context menu option</b></p> <ul style="list-style-type: none"> <li>• <b>Workflow</b> - (  ) <b>Scripts executed by Enterprise Architect's workflow engine</b>; you can create only VB scripts of this type)</li> <li>• <b>Search</b> - (  ) Scripts that can be executed as <b>model searches</b>; these scripts are listed in the Search field of the Model Search window</li> <li>• <b>Diagram</b> - (  ) Scripts that are listed in and can be executed from the Scripts submenu of the <b>Diagram context menu</b></li> <li>• <b>Notes</b> - Your own notes on the script group</li> </ul> |          |

### 18.1.2 Console Tab

#### Topics:

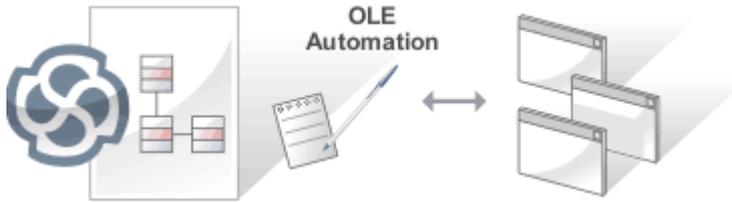
| Topic                   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See also |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Abstract</b>         | <p>The script console is a tab of the <b>Scripting</b> window. It is a command line interpreter with which you can quickly enable a script engine and enter commands to act on the script.</p> <p>You type the commands in the field at the bottom of the tab; when you press the ( <b>Enter</b> ) key, the script console executes the commands and displays any output immediately.</p> <p>You can input two types of command:</p> <ul style="list-style-type: none"> <li>• Console commands</li> <li>• Script commands</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |          |
| <b>Console Commands</b> | <p>Console commands are preceded by the ! character and instruct the console to perform an action. The available console commands are listed below; to list these commands on the <b>Console</b> tab itself (as shown above) type ? in the console field (without the preceding ! character) and press ( <b>Enter</b> ) .</p> <ul style="list-style-type: none"> <li>• <b>c(lear)</b> - clears the console display</li> <li>• <b>sa(ve)</b> - saves the console display to a file</li> <li>• <b>h(elp)</b> - prints a list of commands, as for ?</li> <li>• <b>VB</b> - opens a VBScript console</li> <li>• <b>JA</b> - opens a JavaScript console</li> <li>• <b>JS</b> - opens a JScript console</li> <li>• <b>st(op)</b> - closes any script running console</li> <li>• <b>i(nclude) name</b> - executes the named script item; name is of the format <i>GroupName.ScriptName</i> (spaces are allowed in names)</li> <li>• <b>?</b> - (without the !) lists commands</li> <li>• <b>?name</b> - Outputs the value of a variable name (only if a script console is opened)</li> </ul> |          |
| <b>Script Commands</b>  | <p>A script command is script code that depends on the script engine. Script commands can be executed only once a script console has been created.</p> <p>Examples:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |

| Topic                      | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                            | <p>The following lines, entered into the console, create a <i>VBScript</i> console and then execute the script <i>MyScript</i> in the user group <i>MyGroup</i>.</p> <pre>&gt;! VB &gt;! i MyGr oup. MyScr i pt</pre> <p>The following lines, entered into the console, create a <i>JScript</i> console and then create a variable called <i>x</i> with the value <i>1</i>.</p> <pre>&gt;! JS &gt;var x = 1</pre> <p>The following image shows the result of entering the above <i>JScript</i> example. Remember that you can use <b>?&lt;variable name&gt;</b> to get the current value of any item you have created during the console session.</p>  |          |
| <b>Console Tab Toolbar</b> | <p>The <b>Console</b> tab has two operations available through the toolbar:</p> <ul style="list-style-type: none"> <li>• Open Console (  ) - click on the down-arrow and select to open a VBScript console, JScript console or JavaScript console</li> <li>• Stop Script (  ) - click to stop an executing script and close the current console.</li> </ul>                                                                                                                                                                                                       |          |

**Notes:**

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

## 18.2 Enterprise Architect Object Model



### Topics:

| Topic                                                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also                                                                                                                                                          |
|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Introduction</b>                                  | <p>Automation provides a way for other applications to access the information in an Enterprise Architect model using Windows OLE Automation (ActiveX). Typically this involves scripting clients such as MS Word or Visual Basic, or using scripts created within Enterprise Architect using the <b>Scripting window</b>.</p> <p>The Automation Interface provides a way of accessing the internals of Enterprise Architect models. Examples of things you can do using the Automation Interface include:</p> <ul style="list-style-type: none"> <li>• Perform repetitive tasks, such as update the version number for all elements in a model</li> <li>• Generate code from a State Machine diagram</li> <li>• Produce custom reports</li> <li>• Perform ad hoc queries</li> </ul> | <a href="#">Scripting</a> <sup>[1817]</sup>                                                                                                                       |
| <b>Connecting to the Automation Interface</b>        | All development environments capable of generating <i>ActiveX</i> Com clients should be able to connect to the Enterprise Architect Automation Interface. This guide provides detailed instructions on connecting to the interface using Microsoft Visual Basic 6.0, Borland Delphi 7.0, Microsoft C# and Java. There are also more detailed steps on how to set-up Visual Basic; the principles are applicable to other languages.                                                                                                                                                                                                                                                                                                                                                 | <a href="#">Connecting to the Interface</a> <sup>[1823]</sup><br><a href="#">Set References in Visual Basic</a> <sup>[1825]</sup>                                 |
| <b>Examples and Tips</b>                             | Instruction on how to use the Automation Interface is provided by means of sample code. See pointers to the samples and other available resources. Also, consult the extensive Reference Section.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <a href="#">Pointers to the Samples</a> <sup>[1826]</sup><br><a href="#">Available Resources</a> <sup>[1828]</sup><br><a href="#">Reference</a> <sup>[1828]</sup> |
| <b>Calling Executables from Enterprise Architect</b> | Enterprise Architect can be set up to call an external application. You can pass parameters on the current position selected in the <b>Project Browser</b> to the application being called. For instructions, go to the Call from Enterprise Architect topic. A more sophisticated method is to create Add-Ins, which are discussed in a separate topic.                                                                                                                                                                                                                                                                                                                                                                                                                            | <a href="#">Call from Enterprise Architect</a> <sup>[1827]</sup><br><a href="#">Add-Ins</a> <sup>[1967]</sup>                                                     |

### 18.2.1 Using the Automation Interface

This section provides instructions on how to connect to and use the Automation Interface, including:

| Topic                       | Link                                                          |
|-----------------------------|---------------------------------------------------------------|
| Connecting to the Interface | <a href="#">Connecting to the Interface</a> <sup>[1823]</sup> |

| Topic                              | Link                                               |
|------------------------------------|----------------------------------------------------|
| Setting References In Visual Basic | <a href="#">Setting References In Visual Basic</a> |
| Examples and Tips                  | <a href="#">Examples and Tips</a>                  |

### 18.2.1.1 Connect to the Interface

All development environments capable of generating ActiveX Com clients should be able to connect to the Enterprise Architect Automation Interface.

By way of example, the following sections describe how to connect using several such tools. The procedure might vary slightly with different versions of these products.

#### How To - Microsoft Visual Basic 6.0

| Step | Action                                                                                                                                                                                                                                                             | See Also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Create a new project.                                                                                                                                                                                                                                              |          |
| 2.   | Select the <b>Project   References</b> menu option.                                                                                                                                                                                                                |          |
| 3.   | Select <i>Enterprise Architect Object Model 2.0</i> from the list. (If this does not appear, go to the command line and re-register Enterprise Architect using<br><br><i>EA.exe /unregister</i><br><br>then<br><br><i>EA.exe /register</i> ).                      |          |
| 4.   | See the general library documentation on the use of Classes. The following example creates and opens a repository object:<br><br><pre>Public Sub ShowRepository()<br/>    Dim MyRep As New EA.Repository<br/>    MyRep.OpenFile "c:\eat est.eap"<br/>End Sub</pre> |          |

#### How To - Borland Delphi 7.0

| Step | Action                                                                                                                                                                                                                                        | See Also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Create a new project.                                                                                                                                                                                                                         |          |
| 2.   | Select the <b>Project   Import Type Library</b> menu option.                                                                                                                                                                                  |          |
| 3.   | Select <b>Enterprise Architect Object Model 2.0</b> from the list. (If this does not appear, go to the command line and re-register Enterprise Architect using<br><br><i>EA.exe /unregister</i><br><br>then<br><br><i>EA.exe /register</i> ). |          |

| Step | Action                                                                                                                                                                                                                                                                                                     | See Also |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 4.   | Click on the <b>Create Unit</b> button.                                                                                                                                                                                                                                                                    |          |
| 5.   | Include <i>EA_TLB</i> in Project1's <i>Uses</i> clause.                                                                                                                                                                                                                                                    |          |
| 6.   | See the general library documentation on the use of Classes. The following example creates and opens a repository object: <pre> procedure TForm1.Button1Click(Sender: TObject); var   r: TRepository;   b: boolean; begin   r := TRepository.Create(nil);   b := r.OpenFile('c:\eat est.eap'); end; </pre> |          |

#### How To - Microsoft C#

| Step | Action                                                                                                                                                                                                                                                                                  | See Also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1.   | Select the Visual Studio <b>Project   Add Reference</b> menu option.                                                                                                                                                                                                                    |          |
| 2.   | Click on the <b>Browse</b> tab.                                                                                                                                                                                                                                                         |          |
| 3.   | Navigate to the folder in which you installed Enterprise Architect (usually <i>Program Files/ Sparx Systems/ EA</i> ) and select <i>Interop. EA.dll</i> .                                                                                                                               |          |
| 4.   | See the general library documentation on the use of Classes. The following example creates and opens a repository object: <pre> private void button1_Click(object sender, System.EventArgs e) {   EA.Repository r = new EA.RepositoryClass();   r.OpenFile("c:\\eat est.eap"); } </pre> |          |

#### How To - Java

| Step | Action                                                                                                                                                                                                                                                 | See Also                                                                                                        |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| 1.   | Copy the file <i>SSJavaCOM.dll</i> from the <i>Java API</i> subdirectory of your installed directory (usually <i>Program Files/ Sparx Systems/ EA</i> ) into any location within the Windows PATH. For example, the <i>windows\system32</i> directory. |                                                                                                                 |
| 2.   | Copy the <i>eaapi.jar</i> file from the <i>Java API</i> subdirectory of your installed directory (usually <i>Program Files/ Sparx Systems/ EA</i> ) to a location in the Java CLASSPATH or where the Java class loader can find it at run time.        |                                                                                                                 |
| 3.   | All of the Classes described in the documentation are in the package <i>org.sparx</i> . See the <b>general library documentation</b> for their use. The following example creates and opens a repository object.                                       | <a href="#">Reference</a>  |

| Step | Action                                                                                                                                     | See Also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <pre> public void OpenRepository() {     org.sparx.Repository r = new org.sparx. Repository();     r.OpenFile("c:\\eat est.eap"); } </pre> |          |

### 18.2.1.1.1 Set References In Visual Basic

This topic describes how to use the Enterprise Architect ActiveX interface with Visual Basic (VB). Use is ensured for Visual Basic version 6. This might vary slightly with versions other than version 6.

It is assumed that you have accessed VB through a Microsoft Application such as VB 6.0, MS Word or MS Access. If the code is not called from within Word, the *Word VB* reference must also be set.

On creating a new VB project, set a reference to an Enterprise Architect Type Library and a Word Type Library. Follow the steps below:

#### How To

| Step | Action                                                                                                 | See Also |
|------|--------------------------------------------------------------------------------------------------------|----------|
| 1.   | Select the <b>Tools   References</b> menu option. The following dialog displays:                       |          |
| 2.   | Select the <b>Enterprise Architect Object Model 2.10</b> checkbox from the list.                       |          |
| 3.   | Do the same for VB or VB Word: select the checkbox for the <b>Microsoft Word 10.0 Object Library</b> . |          |
| 4.   | Click on the <b>OK</b> button.                                                                         |          |

#### Notes:

- If Enterprise Architect Object Model 2.10 does not appear in the list, go to the command line and manually re-enter Enterprise Architect using the following:
  - To unregister Enterprise Architect: `ea.exe /unregister`
  - To register Enterprise Architect: `ea.exe /register`
- **Visual Basic 5/6** users should also note that the version number of the Enterprise Architect interface is stored in the VBP project file in a form similar to the following:

```

Reference=* \ G{ 64FB2BF4- 9EFA- 11D2- 8307- C45586000000} #2. 2#0#... \ . . . \ . . .
\ Program Files\ Sparx Systems\ EA\ EA. TLB#Ent erpri se Archi tect Obj ect
Model 2. 02

```

If you experience problems moving from one version of Enterprise Architect to another, open the VBP file in a text editor and remove this line. Then open the project in Visual Basic and use Project-References to create a new reference to the Enterprise Architect Object model

Reference to objects in Enterprise Architect and Word should now be available in the **Object Browser**. This can be accessed from the main menu by selecting **View | Object Browser**, or by pressing ( **F2** )

The drop-down list on the top-left of the window should now include Enterprise Architect and Word. If MS-Project is installed this must also be set up

### 18.2.1.2 Examples and Tips

#### Topics:

| Topic                                   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Usage</b>                            | <p>Instructions for using the interface are provided through sample code. There are several sets of examples:</p> <ul style="list-style-type: none"> <li>• VB 6 and C# examples are available in the Code Samples folder under your Enterprise Architect installation (default: C: \ P r o g r a m F i l e s \ S p a r x S y s t e m s \ E A \ C o d e S a m p l e s )</li> <li>• Enterprise Architect can be set up to <b>call an external application</b></li> <li>• Several VB.NET code snippets are provided in the <b>reference section</b></li> <li>• A comprehensive example of using Visual Basic to create MS Word documentation is available from the internet at <b>www.sparxsystems.com/resources/developers/autint_vb.html</b></li> <li>• Additional samples are available from the Sparx Systems website; see the <b>Available Resources</b> topic</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <p><a href="#">Call an External Application</a> <small>[1827]</small></p> <p><a href="#">Code Samples</a> <small>[1952]</small></p> <p><a href="http://www.sparxsystems.com/resources/developers/autint_vb.html">www.sparxsystems.com/resources/developers/autint_vb.html</a><br/>(Online Resource)</p> <p><a href="#">Available Resources</a> <small>[1828]</small></p> |
| <b>Tips and Tricks</b>                  | <p>Additionally, you should note the following tips and tricks:</p> <ul style="list-style-type: none"> <li>• An instance of the Enterprise Architect (EA.exe) process is executed when you initialize a new repository object. This process must remain running in order to perform automation tasks. If the main window is visible, you can safely minimize it, but it must remain running.</li> <li>• The Enterprise Architect ActiveX Interface is a functional interface rather than a data interface. When you load data through the interface there is a noticeable delay as Enterprise Architect user interface elements (such as Windows, menus) are loaded and the specified database connection is established.</li> <li>• Collections use a zero-based index; for example, Repository.Models(0) represents the first model in the repository.</li> <li>• During the development of your client software your program might terminate unexpectedly and leave EA.exe running in such a state that it is unable to support further interface calls. If your program terminates abnormally, ensure that Enterprise Architect is not left running in the background (see the Windows <b>Task Manager / Process</b> tab).</li> <li>• A handle to a currently running instance of Enterprise Architect can be obtained through the use of a GetObject() call. For more information, refer to the reference page for the <b>App</b> object. Accessing your Enterprise Architect model via the App object enables querying the current User Interface status, such as using GetContextItem() on the <b>Repository</b> object to detect the current selection by the user, allowing for rapid prototyping and testing</li> </ul> | <p><a href="#">Repository Class</a> <small>[1855]</small></p> <p><a href="#">App Class</a> <small>[1830]</small></p>                                                                                                                                                                                                                                                     |
| <b>Enterprise Architect Not Closing</b> | <p>If your automation controller was written using the .NET framework, Enterprise Architect does not close even after you release all your references to it. To force the release of the COM pointers, call the memory management functions as shown</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <p><a href="#">Tricks and Traps</a> <small>[1972]</small></p>                                                                                                                                                                                                                                                                                                            |

| Topic | Detail                                                                                                                                                                                                                                                | See also |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>below:</p> <pre>GC. Collect (); GC. Wait For Pending Finalizers ();</pre> <p>There are additional concerns when controlling a running instance of Enterprise Architect that loads Add-Ins - see the <b>Tricks and Traps</b> topic for details.</p> |          |

### 18.2.1.2.1 Call from Enterprise Architect

Enterprise Architect can be set up to call an external application. You can pass parameters on the current position selected in the **Project Browser** to the application being called.

**Access:** **Tools | Customize > Tools**

To define an application that you can run from Enterprise Architect, select the **Tools | Customize** menu option. The **Customize** dialog displays. Select the **Tools** tab.

**Topics:**

| Topic       | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See also |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Uses</b> | <p>With this you can:</p> <ul style="list-style-type: none"> <li>• Add a command line for an application</li> <li>• Define parameters to pass to this application</li> </ul> <p>The parameters required for running the AutInt executable are:</p> <ul style="list-style-type: none"> <li>• The Enterprise Architect file parameter \$f and</li> <li>• The current PackageID \$p</li> </ul> <p>Hence the arguments should simply contain: \$f , \$p</p> |          |

The available parameters for passing information to external applications are:

| Parameter | Description                                | Notes                                         |
|-----------|--------------------------------------------|-----------------------------------------------|
| \$d       | Diagram ID                                 | ID for accessing associated diagram.          |
| \$D       | Diagram GUID                               | GUID for accessing the associated diagram.    |
| \$e       | Comma separated list of element IDs        | All elements selected in the current diagram. |
| \$E       | Comma separated list of element GUIDs      | All elements selected in the current diagram. |
| \$f       | Project Name                               | For example: C:\projects\EAexample.eap.       |
| \$F       | Calling Application (Enterprise Architect) | 'Enterprise Architect'.                       |
| \$p       | Current Package ID                         | For example: 144.                             |
| \$P       | Package GUID                               | GUID for accessing this package.              |

Once this has been set up, the application can be called from the main menu in Enterprise Architect using the **Add-Ins | <YourApplication>** menu option.

### 18.2.1.2.2 Available Resources

Other available resources include:

| Resource                                                                                                                      | Download Link                                                                                                                                                  |
|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VB 6 Add-In for generating MS Word documentation.                                                                             | <a href="http://www.sparxsystems.com/resources/developers/autint_vb.html">www.sparxsystems.com/resources/developers/autint_vb.html</a>                         |
| VB 6 Add-In to display a custom ActiveX graph control within the Enterprise Architect window as a new view.                   | <a href="http://www.sparxsystems.com/resources/developers/autint_vb_custom_view.html">www.sparxsystems.com/resources/developers/autint_vb_custom_view.html</a> |
| A basic Add-In framework written in C#. Useful as a starting point for authoring your own custom Enterprise Architect Add-In. | <a href="http://www.sparxsystems.com/bin/CS_AddinFramework.zip">www.sparxsystems.com/bin/CS_AddinFramework.zip</a>                                             |
| An extension on the <i>CS_AddinFramework</i> example showing how to export Tagged Values to a .csv file.                      | <a href="http://www.sparxsystems.com/bin/CS_AddinTaggedCSV.zip">www.sparxsystems.com/bin/CS_AddinTaggedCSV.zip</a>                                             |
| A basic Add-In skeleton written in Delphi.                                                                                    | <a href="http://www.sparxsystems.com/bin/DelphiDemo.zip">www.sparxsystems.com/bin/DelphiDemo.zip</a>                                                           |
| A simple example Add-In written in C#.                                                                                        | <a href="http://www.sparxsystems.com/bin/CS_Sample.zip">www.sparxsystems.com/bin/CS_Sample.zip</a>                                                             |

#### Learn More:

- For further information, see [www.sparxsystems.com/resources/developers/autint.html](http://www.sparxsystems.com/resources/developers/autint.html)

## 18.2.2 Reference

This section provides detailed information on all the objects available in the object model provided by the Automation Interface, covering:

| Topic                                | Link                                                                   |
|--------------------------------------|------------------------------------------------------------------------|
| Interface Overview Package           | <a href="#">Interface Overview Package</a> <sup>[1829]</sup>           |
| App Object                           | <a href="#">App Object</a> <sup>[1830]</sup>                           |
| Enumerations                         | <a href="#">Enumerations</a> <sup>[1831]</sup>                         |
| Repository Package                   | <a href="#">Repository Package</a> <sup>[1838]</sup>                   |
| Element Package                      | <a href="#">Element Package</a> <sup>[1873]</sup>                      |
| Element Features Package             | <a href="#">Element Features Package</a> <sup>[1897]</sup>             |
| Connector Package                    | <a href="#">Connector Package</a> <sup>[1910]</sup>                    |
| Diagram Package                      | <a href="#">Diagram Package</a> <sup>[1920]</sup>                      |
| Project Interface Package            | <a href="#">Project Interface Package</a> <sup>[1929]</sup>            |
| Document Generator Interface Package | <a href="#">Document Generator Interface Package</a> <sup>[1947]</sup> |

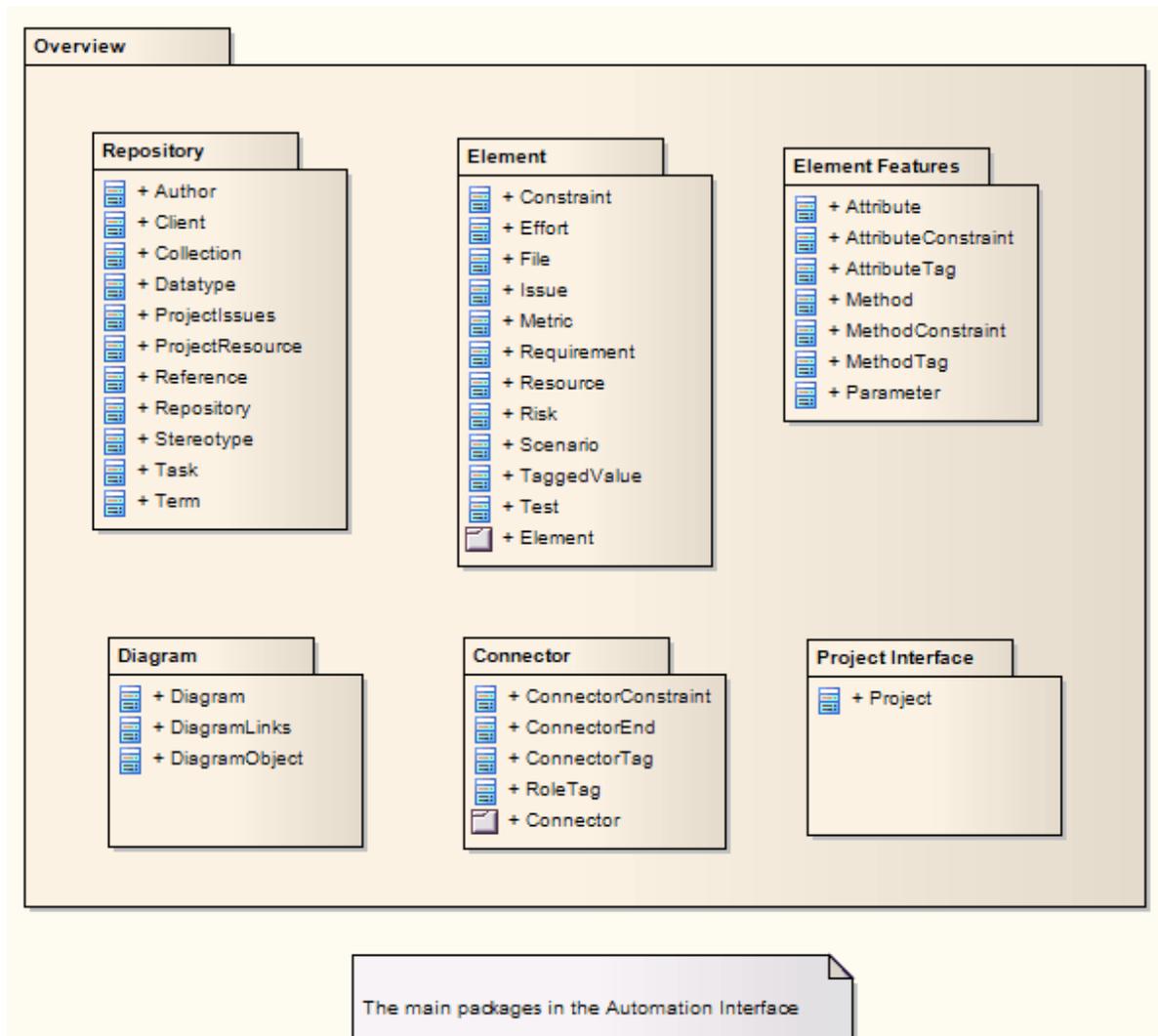
| Topic                  | Link                                                     |
|------------------------|----------------------------------------------------------|
| Mail Interface Package | <a href="#">Mail Interface Package</a> <sup>[1950]</sup> |
| Code Samples           | <a href="#">Code Samples</a> <sup>[1952]</sup>           |

### 18.2.2.1 Interface Overview Package

This package provides an overview of the main components of the Automation Interface. These include the:

| Topic                           | Detail                                                                               | Link                                                       |
|---------------------------------|--------------------------------------------------------------------------------------|------------------------------------------------------------|
| <b>Repository Package</b>       | Represents the model as a whole and provides entry to model packages and collections | <a href="#">Repository Package</a> <sup>[1838]</sup>       |
| <b>Element Package</b>          | Identifies the basic structural units (such as Class, Use Case and Object)           | <a href="#">Element Package</a> <sup>[1873]</sup>          |
| <b>Element Features Package</b> | Identifies the attributes and operations defined on an element                       | <a href="#">Element Features Package</a> <sup>[1897]</sup> |
| <b>Diagram Package</b>          | Describes the visible drawings contained in the model                                | <a href="#">Diagram Package</a> <sup>[1920]</sup>          |
| <b>Connector Package</b>        | Defines the relationships between elements                                           | <a href="#">Connector Package</a> <sup>[1910]</sup>        |

The following diagram illustrates the main interface packages and their associated contents. Each element in this document is creatable by Automation and can be accessed through the various collections that exist or, in some cases, directly.



The *Repository* Class is the starting point for all use of the automation interface. It contains the high level system objects and entry point into the model itself using the *Models* collection and the other system level collections.

### 18.2.2.2 App Object

The *App* object represents a running instance of Enterprise Architect. Its object provides access to the Automation Interface.

| Attribute  | Type       | Notes                                                  |
|------------|------------|--------------------------------------------------------|
| Project    | Project    | Read only. Provides a handle to the Project Interface. |
| Repository | Repository | Read only. Provides a handle to the Repository object. |
| Visible    | Boolean    | Read/Write. Whether or not the application is visible. |

| Topic                      | Detail                                                                                                                          | Link |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------|------|
| <b>GetObject() Support</b> | The <i>App</i> object is creatable and a handle can be obtained by creating one. In addition, clients can use the equivalent of |      |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Link |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|       | <p>Visual Basic's <i>GetObject()</i> to obtain a reference to a currently running instance of Enterprise Architect.</p> <p>Use this method to more quickly test changes to Add-Ins and external clients, as the Enterprise Architect application and data files do not have to be constantly re-loaded.</p> <p>For example:</p> <pre>Dim App as EA.App Set App = GetObject(, "EA.App") MsgBox App.Repository.Models.Count</pre> <p>Another example, which uses the App object without saving it to a variable:</p> <pre>Dim Rep as EA.Repository Set Rep = GetObject(, "EA.App"). Repository MsgBox Rep.ConnectionString</pre> |      |

### 18.2.2.3 Enumerations

The following enumerations are defined by the Automation Interface:

| Enumeration                   | Link                                                   |
|-------------------------------|--------------------------------------------------------|
| Constant Layout Styles        | <a href="#">ConstLayoutStyles</a> <sup>[1832]</sup>    |
| Create Baseline Flag          | <a href="#">CreateBaselineFlag</a> <sup>[1832]</sup>   |
| Create Model Type             | <a href="#">CreateModelType</a> <sup>[1832]</sup>      |
| Document Break                | <a href="#">DocumentBreak</a> <sup>[1833]</sup>        |
| Document Type                 | <a href="#">DocumentType</a> <sup>[1833]</sup>         |
| EA Edition Types              | <a href="#">EAEditionTypes</a> <sup>[1833]</sup>       |
| Enumeration Relation Set Type | <a href="#">EnumRelationSetType</a> <sup>[1834]</sup>  |
| Export Package XML Flag       | <a href="#">ExportPackageXMLFlag</a> <sup>[1834]</sup> |
| Mail Interface Message Flag   | <a href="#">MessageFlag</a> <sup>[1835]</sup>          |
| MDG Menus                     | <a href="#">MDGMenus</a> <sup>[1834]</sup>             |
| Object Type                   | <a href="#">ObjectType</a> <sup>[1835]</sup>           |
| PropType                      | <a href="#">PropType</a> <sup>[1836]</sup>             |
| Reload Type                   | <a href="#">ReloadType</a> <sup>[1837]</sup>           |
| Scenario Diagram Type         | <a href="#">ScenarioDiagramType</a> <sup>[1837]</sup>  |
| Scenario Step Type            | <a href="#">ScenarioStepType</a> <sup>[1838]</sup>     |
| Scenario Test Type            | <a href="#">ScenarioTestType</a> <sup>[1838]</sup>     |
| XMI Type                      | <a href="#">XMIType</a> <sup>[1838]</sup>              |

### 18.2.2.3.1 ConstLayoutStyles

The *enum* values defined here are used exclusively for the *Lay Out a Diagram* method. They enable you to define the layout options as depicted in the **Diagram | Layout Diagram** menu option.

| Value                              | Meaning                                                                     |
|------------------------------------|-----------------------------------------------------------------------------|
| <i>IsCrossReduceAggressive</i>     | Perform aggressive Cross-reduction in the layout process (time consuming)   |
| <i>IsCycleRemoveDFS</i>            | Use the <i>Depth First Cycle Removal</i> algorithm                          |
| <i>IsCycleRemoveGreedy</i>         | Use the <i>Greedy Cycle Removal</i> algorithm                               |
| <i>IsDiagramDefault</i>            | Use existing layout options specified for this diagram                      |
| <i>IsInitializeDFSIn</i>           | Initialize the layout using the <i>Depth First Search Inward</i> algorithm  |
| <i>IsInitializeNaive</i>           | Initialize the layout using the <i>Naïve Initialize Indices</i> algorithm   |
| <i>IsInitializeDFSOut</i>          | Initialize the layout using the <i>Depth First Search Outward</i> algorithm |
| <i>IsLayeringLongestPathSink</i>   | Layer the diagram using the <i>Longest Path Sink</i> algorithm              |
| <i>IsLayeringLongestPathSource</i> | Layer the diagram using the <i>Longest Path Source</i> algorithm            |
| <i>IsLayeringOptimalLinkLength</i> | Layer the diagram using the <i>Optimal Link Length</i> algorithm            |
| <i>IsLayoutDirectionDown</i>       | Direct connectors to point down                                             |
| <i>IsLayoutDirectionLeft</i>       | Direct connectors to point left                                             |
| <i>IsLayoutDirectionRight</i>      | Direct connectors to point right                                            |
| <i>IsLayoutDirectionUp</i>         | Direct connectors to point up                                               |
| <i>IsProgramDefault</i>            | Use factory default layout options as specified by Enterprise Architect     |

#### Learn More:

- [Lay Out a Diagram](#) <sup>[619]</sup>

### 18.2.2.3.2 CreateBaselineFlag

The *CreateBaselineFlag* enumeration is used in Baseline Management, when creating a Baseline.

| Value               | Meaning                                                                                        |
|---------------------|------------------------------------------------------------------------------------------------|
| <i>cbSaveToStub</i> | Baseline this package with only immediate children (child packages are included as stubs only) |

### 18.2.2.3.3 CreateModelType

The *CreateModelType* enumeration is used in the **CreateModel** method on the Repository Class.

| Value                | Meaning                                                           |
|----------------------|-------------------------------------------------------------------|
| <i>cmEAPFromBase</i> | Create a copy of the EABase model file to the specified file path |

| Value                         | Meaning                                                                                                             |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <i>cmEAPFromSQLRepository</i> | Create a .eap file shortcut to an SQL-based repository; requires user interaction to provide sql connection details |

**Learn More:**

- [CreateModel Method](#)<sup>[1859]</sup>

**18.2.2.3.4 DocumentBreak**

The *DocumentBreak* enumeration is used in the **InsertBreak** method on the *DocumentGenerator* Class.

| Value               | Meaning                                |
|---------------------|----------------------------------------|
| <i>breakPage</i>    | Insert a page break in the document    |
| <i>breakSection</i> | Insert a section break in the document |

**Learn More:**

- [InsertBreak Method](#)<sup>[1949]</sup>

**18.2.2.3.5 DocumentType**

The *DocumentType* enumeration is used in the **SaveDocument** method on the *DocumentGenerator* Class.

| Value         | Meaning                                            |
|---------------|----------------------------------------------------|
| <i>dtHTML</i> | Save the document file to disk as an HTML document |
| <i>dtRTF</i>  | Save the document file to disk as an RTF document  |

**Learn More:**

- [SaveDocument](#)<sup>[1950]</sup>

**18.2.2.3.6 EAEditionTypes**

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b> | <p>The <i>EAEditionTypes</i> enumeration identifies the current level of licensed functionality available; for example:</p> <pre>EAEditionTypes theEdition = theRepository.Get EAEdition(); if ( theEdition == EAEditionTypes. pi Desktop ) ... else if ( theEdition == EAEditionTypes.pi Professional ) ...</pre> <p>The enumeration defines the following formal values:</p> <ul style="list-style-type: none"> <li>• <i>piLite</i></li> <li>• <i>piDesktop</i></li> <li>• <i>piProfessional</i></li> </ul> |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                         | See Also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <ul style="list-style-type: none"> <li>• <i>piCorporate</i></li> <li>• <i>piBusiness</i></li> <li>• <i>piSystemEng</i></li> <li>• <i>piUltimate</i></li> </ul> <p>There is no separate value for the trial edition; the <i>Repository.GetEAEdition()</i> function returns the appropriate <b>EAEditionTypes</b> value for whichever edition the user has selected to trial</p> |          |

### 18.2.2.3.7 EnumRelationSetType

This enumeration represents values returned from the *GetRelationSet* method of the **Element** object.

| Value                    | Meaning                                                      |
|--------------------------|--------------------------------------------------------------|
| <i>rsDependEnd</i>       | List of elements that depend on the current element          |
| <i>rsDependStart</i>     | List of elements that the current element depends on         |
| <i>rsGeneralizeEnd</i>   | List of elements that are generalized by the current element |
| <i>rsGeneralizeStart</i> | List of elements that the current element generalizes        |
| <i>rsParents</i>         | List of all parent elements of the current element           |
| <i>rsRealizeEnd</i>      | List of elements that are realized by the current element    |
| <i>rsRealizeStart</i>    | List of elements that the current element realizes           |

#### Learn More:

- [Element Class](#) <sup>1877</sup>

### 18.2.2.3.8 ExportPackageXMIFlag

The *ExportPackageXMIFlag* enumeration is used in package control, when exporting to XML.

| Value               | Meaning                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------|
| <i>epSaveToStub</i> | Export this package with only immediate children (child packages are included as stubs only) |

### 18.2.2.3.9 MDGMenus

Use this enumeration when providing the *HiddenMenus* property to *MDG\_GetProperty*.

These options are exclusive of one another and can be read or added to hide more than one menu.

| Value                 | Meaning                               |
|-----------------------|---------------------------------------|
| <i>mgBuildProject</i> | Hide <b>Build Project</b> menu option |
| <i>mgMerge</i>        | Hide <b>Merge</b> menu option         |

| Value        | Meaning                     |
|--------------|-----------------------------|
| <i>mgRun</i> | Hide <b>Run</b> menu option |

**Learn More:**

- [MDG.GetProperty](#)<sup>[2046]</sup>

**18.2.2.3.10 MessageFlag**

The *MessageFlag* enumeration is used in both the **SendMessage** and **ComposeMailMessage** methods of the MailInterface, to specify a flag to attach to the message.

| Value             | Meaning                               |
|-------------------|---------------------------------------|
| <b>mfNone</b>     | Do not flag the message               |
| <b>mfComplete</b> | Flag the message as 'Complete'        |
| <b>mfPurple</b>   | Flag the message with a 'Purple' flag |
| <b>mfOrange</b>   | Flag the message with a 'Orange' flag |
| <b>mfGreen</b>    | Flag the message with a 'Green' flag  |
| <b>mfYellow</b>   | Flag the message with a 'Yellow' flag |
| <b>mfBlue</b>     | Flag the message with a 'Blue' flag   |
| <b>mfRed</b>      | Flag the message with a 'Red' flag    |

**18.2.2.3.11 ObjectType**

| Topic                           | Detail                                                                                                                                                                                                                                                                                               | See Also |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b>            | <p>The <i>ObjectType</i> enumeration identifies Enterprise Architect object types even when referenced through a Dispatch interface</p> <p>For example:</p> <pre> Object ob = Repository.GetElementById ( 13 ); if ( ob.ObjectType == otElement ) ; else if ( ob.ObjectType == otAuthor ) ... </pre> |          |
| <b>Valid Enumeration Values</b> | <p><i>otAttribute</i><br/> <i>otAttributeConstraint</i><br/> <i>otAttributeTag</i><br/> <i>otAuthor</i><br/> <i>otClient</i><br/> <i>otCollection</i><br/> <i>otConnector</i><br/> <i>otConnectorConstraint</i><br/> <i>otConnectorEnd</i><br/> <i>otConnectorTag</i><br/> <i>otConstraint</i></p>   |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See Also |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <i>otCustomProperty</i><br><i>otDatatype</i><br><i>otDiagram</i><br><i>otDiagramLink</i><br><i>otDiagramObject</i><br><i>otEffort</i><br><i>otElement</i><br><i>otEventProperties</i><br><i>otEventProperty</i><br><i>otFile</i><br><i>otIssue</i><br><i>otMailInterface</i><br><i>otMethod</i><br><i>otMethodConstraint</i><br><i>otMethodTag</i><br><i>otMetric</i><br><i>otModel</i><br><i>otNone</i><br><i>otPackage</i><br><i>otParameter</i><br><i>otParamTag</i><br><i>otPartition</i><br><i>otProject</i><br><i>otProjectIssues</i><br><i>otProjectResource</i><br><i>otProperties</i><br><i>otProperty</i><br><i>otPropertyType</i><br><i>otReference</i><br><i>otRepository</i><br><i>otRequirement</i><br><i>otResource</i><br><i>otRisk</i><br><i>otRoleTag</i><br><i>otScenario</i><br><i>otScenarioExtension</i><br><i>otScenarioStep</i><br><i>otStereotype</i><br><i>otSwimlane</i><br><i>otSwimlaneDef</i><br><i>otSwimlanes</i><br><i>otTaggedValue</i><br><i>otTask</i><br><i>otTerm</i><br><i>otTest</i><br><i>otTransition</i> |          |

#### 18.2.2.3.12 PropType

The *PropType* enumeration gives the automation programmer an indication of what sort of data is going to be stored by this property.

| Value          | Meaning                                |
|----------------|----------------------------------------|
| <i>ptArray</i> | An array containing values of any type |

| Value                  | Meaning                                                                                                    |
|------------------------|------------------------------------------------------------------------------------------------------------|
| <i>ptBoolean</i>       | True or False                                                                                              |
| <i>ptEnum</i>          | A string being an entry in the semi-colon separated list specified in the validation field of the Property |
| <i>ptFloatingPoint</i> | 4 or 8 byte floating point value                                                                           |
| <i>ptInteger</i>       | 16-bit or 32-bit signed integer                                                                            |
| <i>ptString</i>        | Unicode string                                                                                             |

### 18.2.2.3.13 ReloadType

This enumeration represents values returned from the *GetReloadItem* and *PeekReloadItem* methods of the *ModelWatcher* Class. It has four possible values, which define the type of change that was made to a model.

| Value                | Meaning                                                                         |
|----------------------|---------------------------------------------------------------------------------|
| <i>rtElement</i>     | The <i>Item</i> parameter represents a particular element that must be reloaded |
| <i>rtEntireModel</i> | Entire model must be reloaded to ensure that all changes are reloaded           |
| <i>rtNone</i>        | No change in the model                                                          |
| <i>rtPackage</i>     | The <i>Item</i> parameter represents a particular package that must be reloaded |

### 18.2.2.3.14 ScenarioDiagramType

The *ScenarioDiagramType* enumeration provides the following enumeration values to the **Project.GenerateDiagramFromScenario()** method. They specify the type of diagram to generate.

| Value                                  | Meaning                                                | See also                                                         |
|----------------------------------------|--------------------------------------------------------|------------------------------------------------------------------|
| <i>sdActivity</i>                      | Generate an Activity diagram                           | <a href="#">Generated Activity diagram</a> <sup>[677]</sup>      |
| <i>sdActivityWithAction</i>            | Generate an Activity diagram with an Action            | <a href="#">Generated Activity diagram</a> <sup>[677]</sup>      |
| <i>sdActivityWithActionPin</i>         | Generate an Activity diagram with an ActionPin         | <a href="#">Generated Activity diagram</a> <sup>[677]</sup>      |
| <i>sdActivityWithActivityParameter</i> | Generate an Activity diagram with an ActivityParameter | <a href="#">Generated Activity diagram</a> <sup>[677]</sup>      |
| <i>sdRobustness</i>                    | Generate a Robustness diagram                          | <a href="#">Generated Robustness diagram</a> <sup>[683]</sup>    |
| <i>sdRuleFlow</i>                      | Generate a RuleFlow diagram                            | <a href="#">Generated RuleFlow diagram</a> <sup>[678]</sup>      |
| <i>sdSequence</i>                      | Generate a Sequence diagram                            | <a href="#">Generated Sequence diagram</a> <sup>[681]</sup>      |
| <i>sdState</i>                         | Generate a State Machine diagram                       | <a href="#">Generated State Machine diagram</a> <sup>[679]</sup> |

**Learn More:**

- [Project.GenerateDiagramFromScenario\(\) Method](#)<sup>[1935]</sup>

### 18.2.2.3.15 ScenarioStepType

The *ScenarioStepType* enumeration is used to identify the **steps** of a scenario, and the entity performing the step.

| Value           | Meaning                                                     | See also |
|-----------------|-------------------------------------------------------------|----------|
| <i>stActor</i>  | Identify that the step is an action performed by an actor   |          |
| <i>stSystem</i> | Identify that the step is an action performed by the system |          |

#### Learn More:

- [Scenario Step Class](#)<sup>[1894]</sup>

### 18.2.2.3.16 ScenarioTestType

The *ScenarioTestType* enumeration provides the following enumeration values to the **Project.GenerateTestFromScenario()** method. They specify the type of test to generate.

| Value             | Meaning                                | See also |
|-------------------|----------------------------------------|----------|
| <i>stExternal</i> | Generate an external Test Case element |          |
| <i>stInternal</i> | Generate an internal test              |          |

#### Learn More:

- [Project.GenerateTestFromScenario\(\) Method](#)<sup>[1937]</sup>
- [Generate Test Classes](#)<sup>[685]</sup>

### 18.2.2.3.17 XMIType

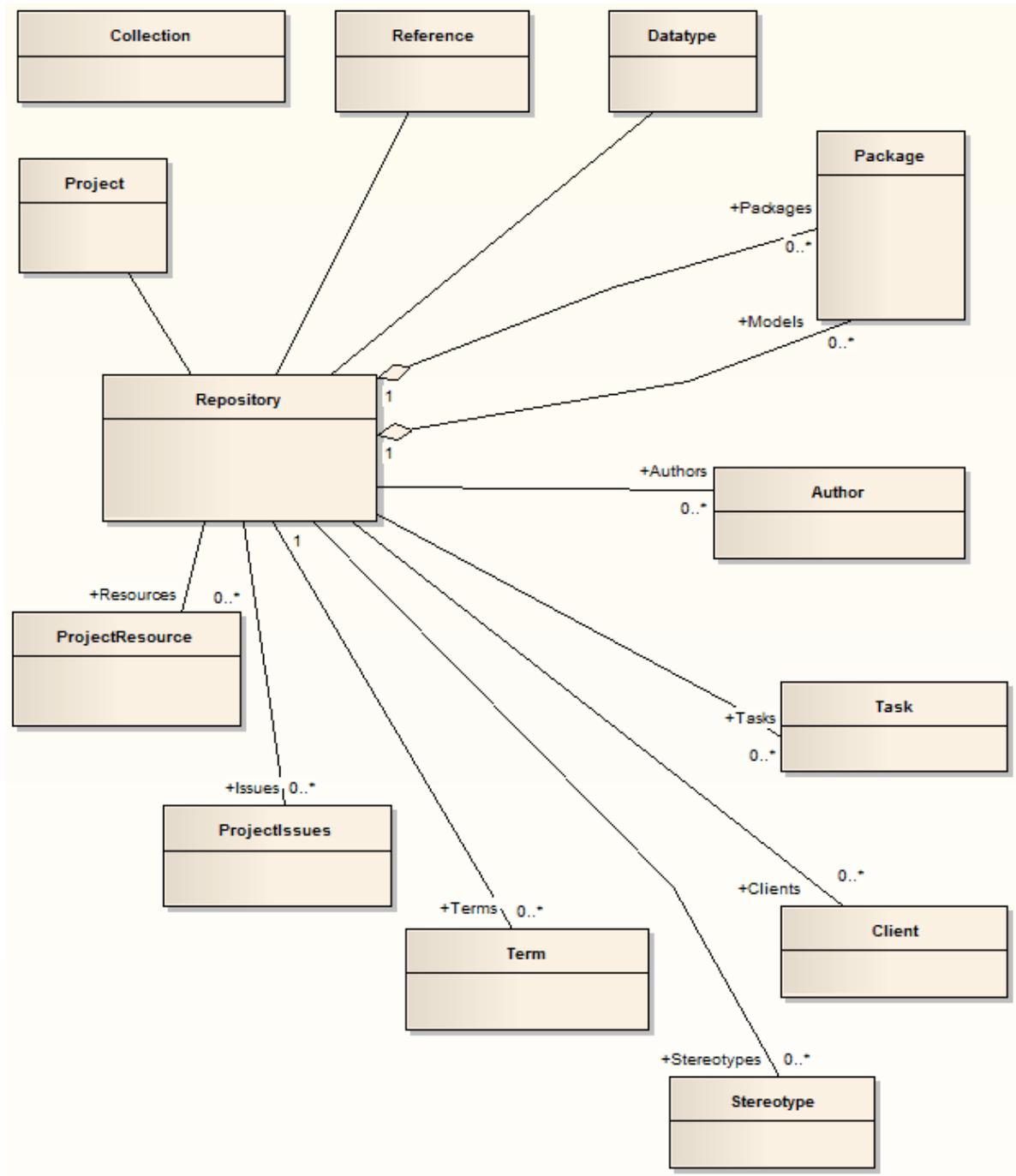
The following enumeration values are used in the *Project.ExportPackageXML()* method. They enable specification of the XML export type.

*xmiEADefault*  
*xmiRoseDefault*  
*xmiEA10*  
*xmiEA11*  
*xmiEA12*  
*xmiEA20*  
*xmiEA21*  
*xmiMOF13*  
*xmiMOF14*  
*xmiRose10*  
*xmiRose11*  
*xmiRose12*

## 18.2.2.4 Repository Package

The *Repository* package contains the high level system objects and entry point into the model itself using the *Models* collection and the other system level collections.

This diagram illustrates the **Repository** and its first level functions and collections.



#### Learn More:

- [Repository Class](#) 1855

#### 18.2.2.4.1 Author Class

An *Author* object represents a named model author. It can be accessed using the Repository *Authors* collection.

Associated table in .EAP file: *t\_authors*

Author Attributes

| Attribute  | Type                                 | Notes                                                                     |
|------------|--------------------------------------|---------------------------------------------------------------------------|
| Name       | String                               | Read/Write. Author name.                                                  |
| Notes      | String                               | Read/Write. Notes about the author.                                       |
| ObjectType | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface. |
| Roles      | String                               | Read/Write. Roles the author might play in this project.                  |

Author Methods

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Updates the current Author object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.               |

**18.2.2.4.2 Client Class**

A Client represents one or more people or organizations related to the project. Accessed using the Repository *Clients* collection.

Associated table in .EAP file: *t\_clients*

Client Attributes

| Attribute    | Type                                 | Notes                                                                              |
|--------------|--------------------------------------|------------------------------------------------------------------------------------|
| EMail        | String                               | Read/Write. EMail address.                                                         |
| Fax          | String                               | Read/Write. Fax number.                                                            |
| Mobile       | String                               | Read/Write. Mobile phone if available.                                             |
| Name         | String                               | Read/Write. Client name.                                                           |
| Notes        | String                               | Read/Write. Notes about client.                                                    |
| ObjectType   | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through the <i>Dispatch</i> interface. |
| Organization | String                               | Read/Write. Associated organization.                                               |
| Phone1       | String                               | Read/Write. Main phone number.                                                     |
| Phone2       | String                               | Read/Write. Second phone number.                                                   |

| Attribute | Type   | Notes                                                    |
|-----------|--------|----------------------------------------------------------|
| Roles     | String | Read/Write. Roles this client might play in the project. |

#### Client Methods

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Updates the current Client object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.               |

#### 18.2.2.4.3 Collection Class

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Link |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| General Usage | <p>This is the main collection Class used by all elements within the Automation Interface. It contains methods to iterate through the collection, refresh the collection and delete an item from the collection. It is important to realize that when <i>AddNew</i> is called, the item is not automatically added to the current collection. The typical steps are:</p> <ul style="list-style-type: none"> <li>• Call <i>AddNew</i> to add a new item</li> <li>• Modify the item as required</li> <li>• Call <i>Update</i> on the item to save it to the database</li> <li>• Call <i>Refresh</i> on the collection to include it in the current set</li> </ul> <p>Delete is much the same; until <i>Refresh</i> is called, the collection still contains a reference to the deleted item, which should not be called.</p> <p>Each can be used to iterate through the collection for languages that support this type of construct.</p> |      |

#### Collection Attributes

| Attribute  | Type                                                | Notes                                                                     |
|------------|-----------------------------------------------------|---------------------------------------------------------------------------|
| Count      | Short                                               | Read only. The number of objects referenced by this list.                 |
| ObjectType | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface. |

#### Collection Methods

| Method                                         | Type          | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See Also                                                                                                                                                                                    |
|------------------------------------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AddNew (string Name, string Type)</b>       | <i>Object</i> | <p>Adds a new item to the current collection.</p> <p>Note that the interface is the same for all collections; you must provide a <i>Name</i> and <i>Type</i> argument. What these are used for depends on the actual collection member. Also note that you must call <i>Update()</i> on the returned object to complete the <i>AddNew</i>. If <i>Update()</i> is not called the object is left in an indeterminate state.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Name: String</li> <li>• Type: String (up to 30 characters long)</li> </ul> |                                                                                                                                                                                             |
| <b>Delete (short index)</b>                    | <i>Void</i>   | <p>Deletes the item at the selected reference.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                             |
| <b>DeleteAt (short index, boolean Refresh)</b> | <i>Void</i>   | <p>Deletes the item at the selected index. The second parameter is currently unused.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> <li>• Refresh: Boolean</li> </ul>                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                             |
| <b>GetAt (short index)</b>                     | <i>Object</i> | <p>Retrieves the array object using a numerical index. If the index is out of bounds, an error occurs.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> </ul>                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                             |
| <b>GetByName (string Name)</b>                 | <i>Object</i> | <p>Gets an item in the current collection by Name.</p> <p>If the collection does not contain any items, the method returns a null value. If the collection contains items, but it was unable to find an object with the specified name, the method raises an exception.</p> <p>Only supported for the following collections: <b>Model</b>, <b>Package</b>, <b>Element</b>, <b>Diagram</b>, and element <b>TaggedValue</b>.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Name: String</li> </ul>                                                   | <p><a href="#">Package Class</a><br/>[1845]</p> <p><a href="#">Element Class</a> [1877]</p> <p><a href="#">Diagram Class</a><br/>[1921]</p> <p><a href="#">TaggedValue Class</a> [1895]</p> |
| <b>GetLastError ()</b>                         | <i>String</i> | <p>Returns a string value describing the most recent error that occurred in relation to this object.</p> <p>This function is rarely used as an exception is thrown when an error occurs.</p>                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                             |
| <b>Refresh ()</b>                              | <i>Void</i>   | <p>Refreshes the collection by re-querying the model and reloading the collection. Should be called after adding a new item or after deleting an item.</p>                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                             |

#### 18.2.2.4.4 Datatype Class

A *Datatype* is a named type that can be associated with attribute or method types. It typically is related to either code engineering or database modeling. Datatypes also indicate which language or database system they relate to. Accessed using the Repository *Datatypes* collection.

Associated table in .EAP file: *t\_datatypes*

##### Datatype Attributes

| Attribute           | Type                                                | Notes                                                                                                                                                                                                                                       |
|---------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DatatypeID</b>   | <i>Long</i>                                         | Read/Write. Instance ID for this datatype within the current model. System maintained.                                                                                                                                                      |
| <b>DefaultLen</b>   | <i>Long</i>                                         | Read/Write. Default length (DDL only).                                                                                                                                                                                                      |
| <b>DefaultPrec</b>  | <i>Long</i>                                         | Read/Write. Default precision (DDL only).                                                                                                                                                                                                   |
| <b>DefaultScale</b> | <i>Long</i>                                         | Read/Write. Default scale (DDL only).                                                                                                                                                                                                       |
| <b>GenericType</b>  | <i>String</i>                                       | Read/Write. The associated generic type for this data type.                                                                                                                                                                                 |
| <b>HasLength</b>    | <i>String</i>                                       | Read/Write. Indicates datatype has a length component.                                                                                                                                                                                      |
| <b>MaxLen</b>       | <i>Long</i>                                         | Read/Write. Maximum length (DDL only).                                                                                                                                                                                                      |
| <b>MaxPrec</b>      | <i>Long</i>                                         | Read/Write. Maximum precision (DDL only).                                                                                                                                                                                                   |
| <b>MaxScale</b>     | <i>Long</i>                                         | Read/Write. Maximum scale (DDL only).                                                                                                                                                                                                       |
| <b>Name</b>         | <i>String</i>                                       | Read/Write. The datatype name (such as <i>integer</i> ). This appears in the related drop-down datatype lists where appropriate.                                                                                                            |
| <b>ObjectType</b>   | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                   |
| <b>Product</b>      | <i>String</i>                                       | Read/Write. The datatype product, such as Java, C++, Oracle.                                                                                                                                                                                |
| <b>Size</b>         | <i>Long</i>                                         | Read/Write. The datatype size.                                                                                                                                                                                                              |
| <b>Type</b>         | <i>String</i>                                       | Read/Write. The type can be <i>DDL</i> for database datatype or <i>Code</i> for language datatypes.                                                                                                                                         |
| <b>UserDefined</b>  | <i>Long</i>                                         | Read/Write. Indicates if datatype is a user defined type or system generated.<br><br>Datatypes distributed with Enterprise Architect are all system generated. Datatypes created in the <b>Datatype</b> dialog are marked <b>1 (true)</b> . |

##### Datatype Methods

| Method                 | Type          | Notes                                                                                                                                                                                 |
|------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |

| Method    | Type    | Notes                                                                                                                                                                        |
|-----------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Update () | Boolean | Updates the current Datatype object after modification or appending a new item.<br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 18.2.2.4.5 EventProperties Class

An *EventProperties* object is passed to *BroadcastFunctions* to facilitate parameter passing.

##### EventProperties Attributes

| Attribute  | Type                                             | Notes                                                                     |
|------------|--------------------------------------------------|---------------------------------------------------------------------------|
| Count      | Long                                             | Read only. Number of parameters being passed to this broadcast event.     |
| ObjectType | <a href="#">ObjectType</a> <small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface. |

##### EventProperties Methods

| Method             | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Get (object Index) | <a href="#">EventProperty</a> <small>[1844]</small> | Read only. Returns an <i>EventProperty</i> in the list, raising an error if <i>Index</i> is out of range.<br>Parameters: <ul style="list-style-type: none"> <li>Index: Variant - can either be a number representing a zero-based index into the array, or a string representing the name of the <i>EventProperty</i>. For example, <i>Props.Get(3)</i> or <i>Props.Get("ObjectID")</i>.</li> </ul> |

#### 18.2.2.4.6 EventProperty Class

*EventProperty* objects are always part of an **EventProperties** collection, and are passed to Add-In methods responding to **broadcast events**.

##### EventProperty Attributes

| Attribute   | Type                                             | Notes                                                                 |
|-------------|--------------------------------------------------|-----------------------------------------------------------------------|
| Description | String                                           | Explanation of what this property represents.                         |
| Name        | String                                           | A string distinguishing this property from others in the list.        |
| ObjectType  | <a href="#">ObjectType</a> <small>[1835]</small> | Distinguishes objects referenced through a Dispatch interface.        |
| Value       | Variant                                          | A string, number or object reference representing the property value. |

[Learn More:](#)

- [EventProperties Class](#)<sup>[1844]</sup>
- [Broadcast Events](#)<sup>[1983]</sup>

#### 18.2.2.4.7 ModelWatcher Class

The *ModelWatcher* object enables an automation client to track changes in a particular model.

##### ModelWatcher Attributes

| Attribute  | Type                                         | Notes                                                                     |
|------------|----------------------------------------------|---------------------------------------------------------------------------|
| ObjectType | <a href="#">ObjectType</a> <sup>[1835]</sup> | Read only. Distinguishes objects referenced through a Dispatch interface. |

##### ModelWatcher Methods

| Methods                               | Type                                         | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetReloadItem</b><br>(object Item) | <a href="#">ReloadType</a> <sup>[1837]</sup> | <p>The object that must be reloaded in order to see all changes is returned through the <i>Item</i> parameter. If there are no changes or the entire model must be reloaded, this value is returned as <b>null</b> (C#) or <b>Nothing</b> (VB).</p> <p>Calling this method clears the records so that the next time it is called the return values refer only to new changes. Returns a value from the <i>ReloadType</i> enumeration that specifies which type of change, if any, has occurred.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Item: Object</li> </ul> |
| <b>PeekReloadItem</b>                 | <a href="#">ReloadType</a> <sup>[1837]</sup> | This method behaves identically to <i>GetReloadItem()</i> but does not clear the change record.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

##### Notes:

- After your model has been loaded, you only create the ModelWatcher once. If you **reload** the model, or load another model, the created ModelWatcher is still valid.

##### Learn More:

- [Refresh View of Shared Project](#)<sup>[187]</sup>

#### 18.2.2.4.8 Package Class

A *Package* object corresponds to a Package element in the Enterprise Architect Project Browser. It is accessed either through the *Repository Models* collection (a Model is a special form of Package) or through the Package *Packages* collection. Note that a Package has an Element object as an attribute; this corresponds to an Enterprise Architect Package element in the *t\_object* table and is used to associate additional information (such as scenarios and constraints) with the logical package. To set additional information for a package, reference the Element object directly. Also note that if you add a Package to a diagram, you should add an instance of the element (not the Package itself) to the *Diagram Object Class* for a diagram.

Associated table in .EAP file: *t\_package*

##### Package Attributes

| Attribute           | Type                                                | Notes                                                                                                                                                                |
|---------------------|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alias               | String                                              | Read only<br>Alias                                                                                                                                                   |
| BatchLoad           | Long                                                | Read/Write<br>Flag to indicate that the package is batch loaded during batch import from controlled packages<br><b>Not currently used</b>                            |
| BatchSave           | Long                                                | Read/Write<br>Boolean value to indicate whether the package is included in the batch XML export list or not.                                                         |
| CodePath            | String                                              | Read/Write. The path to where associated source code is found. Not currently used.                                                                                   |
| Connectors          | <a href="#">Collection</a><br><small>[1841]</small> | Read only. Collection of connectors.                                                                                                                                 |
| Created             | Date                                                | Read/Write. Date the package was created.                                                                                                                            |
| Diagrams            | <a href="#">Collection</a><br><small>[1841]</small> | Read only. A collection of diagrams contained in this package.                                                                                                       |
| Element             | <a href="#">Element</a> <sup>[1877]</sup>           | Read only. The associated element object. Use to get/set common information such as Stereotype, Complexity, Alias, Author, Constraints, Tagged Values and Scenarios. |
| Elements            | <a href="#">Collection</a><br><small>[1841]</small> | Read only. A collection of elements that belong to this package.                                                                                                     |
| Flags               | String                                              | Read/Write. Extended information about the package.                                                                                                                  |
| IsControlled        | Boolean                                             | Read/Write. Indicates if the package has been marked as <i>Controlled</i> .                                                                                          |
| IsModel             | Boolean                                             | Read only. Indicates if the package is a model or a package.                                                                                                         |
| IsNamespace         | Boolean                                             | Read/Write. <b>True</b> is 'package is a Namespace root'. Use <b>0</b> and <b>1</b> to set <b>False</b> and <b>True</b> .                                            |
| IsProtected         | Boolean                                             | Read/Write. Indicates if the package has been marked as <i>Protected</i> .                                                                                           |
| IsVersionControlled | Boolean                                             | Read. Indicates whether or not this package is under version control.                                                                                                |
| LastLoadDate        | Date                                                | Read/Write. The date XML was last loaded for the package.                                                                                                            |
| LastSaveDate        | Date                                                | Read/Write. The date XML was last saved from the package.                                                                                                            |
| LogXML              | Boolean                                             | Read/Write. Indicates if XML export information is to be logged.                                                                                                     |
| Modified            | Date                                                | Read/Write. Date the package was last modified.                                                                                                                      |
| Name                | String                                              | Read/Write. The name of the package.                                                                                                                                 |
| Notes               | String                                              | Read/Write. Notes about this package.                                                                                                                                |
| ObjectType          | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                            |

| Attribute   | Type                                 | Notes                                                                                                                                     |
|-------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Owner       | String                               | Read/Write. The package owner when using controlled packages.                                                                             |
| PackageGUID | Variant                              | Read only. The global Package ID. Valid across models.                                                                                    |
| PackageID   | Long                                 | Read only. The local Package ID number. Valid only in this model file.                                                                    |
| Packages    | <a href="#">Collection</a><br>[1847] | Read only. A collection of contained packages that can be walked through.                                                                 |
| ParentID    | Long                                 | Read/Write. The ID of the package that is the parent of this one. 0 indicates this package is a <i>model</i> (that is, it has no parent). |
| TreePos     | Long                                 | Read/Write. The relative position in the tree compared to other packages (use to sort packages).                                          |
| UMLVersion  | String                               | Read/Write. The UML version for XML export purposes.                                                                                      |
| UseDTD      | Boolean                              | Read/Write. Indicates if a DTD is to be used when exporting XML.                                                                          |
| Version     | String                               | Read/Write. The version of the package.                                                                                                   |
| XMLPath     | String                               | Read/Write. The path to where the XML is saved when using controlled packages.                                                            |

### Package Methods

| Method                             | Type       | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ApplyGroupLock (string aGroupName) | Boolean    | Applies a group lock to the package object, for the specified group, on behalf of the current user.<br><br>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.<br><br>Parameter: <ul style="list-style-type: none"> <li>aGroupName: String - The name of the security group for which to apply the lock.</li> </ul>                                                                    |
| ApplyUserLock ()                   | Boolean    | Applies a user lock to the package object for the current user.<br><br>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                             |
| Clone                              | LDISPATCH  | Inserts a copy of the package into the same parent as the original package.<br><br>Returns the newly-created package.                                                                                                                                                                                                                                                                                                                   |
| FindObject (string DottedID)       | LPDISPATCH | Returns a package, element, attribute or operation matching the parameter <i>DottedID</i> . If the <i>DottedID</i> is not found, an error is returned: <i>Can't find matching object</i> .<br><br>Parameter: <ul style="list-style-type: none"> <li>DottedID: String - Is in the form <i>object.object.object</i> where <i>object</i> is replaced by the name of a package, element attribute or operation. Examples include</li> </ul> |

| Method                                                                                               | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                      |                | MyNamespace.Class1, CStudent.m_Name, MathClass.DoubleIt(int).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>GetLastError ()</b>                                                                               | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>ReleaseUserLock ()</b>                                                                            | <i>Boolean</i> | Removes an existing User or Group lock from the package object.<br><br>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Update ()</b>                                                                                     | <i>Boolean</i> | Update the current package object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.<br><br>Note that a package object also has an <i>element</i> component that must be taken into account. The package object contains information about the package attributes such as hierarchy or contents. The element attribute contains information about, for example, Stereotype, Constraints or Files - all the attributes of a typical element.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>VersionControlAdd (string ConfigGuid, string XMLFile, string Comment, boolean KeepCheckedOut)</b> | <i>Void</i>    | Places the package under version control, using the specified Version Control Configuration and the specified XML filename.<br><br>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.<br><br>It is recommended that the package be saved using <i>Update()</i> before calling <i>VersionControlAdd()</i> , so that any outstanding changes are not lost.<br><br>Parameters: <ul style="list-style-type: none"> <li>• ConfigGuid: String - Name corresponding to the Unique ID of the version control configuration to use.</li> <li>• XMLFile: String - Name of the XML file to use for this package. This filename is relative to the Working Copy folder specified for the Config.</li> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable).</li> <li>• KeepCheckedOut: Boolean - Specify <b>True</b> to add to version control and keep package checked-out.</li> </ul> |
| <b>VersionControlCheckin (string Comment)</b>                                                        | <i>Void</i>    | Perform checkin of the version controlled package (also see <i>VersionControlCheckinEx</i> , below)<br><br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information<br><br>Parameters: <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>VersionControlCheckinEx (string Comment, boolean PreserveCrossPkgRefs)</b>                        | <i>Void</i>    | Perform checkin of the version controlled package<br><br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| Method                                                  | Type | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                         |      | <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> <li>• PreserveCrossPkgRefs: Boolean - Flag to indicate whether to preserve or discard pre-existing Cross Package References when checking-in; this parameter overrides the setting in the Options dialog, XML Specifications page</li> </ul> <p>Unsatisfied cross package references are preserved or discarded according to this setting, without prompting the user; see <b>Learn More</b> below</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>VersionControlCheckout</b><br>(string Comment)       | Void | <p>Perform checkout of the version controlled package</p> <p>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> </ul> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke the <b>Repository.ScanXMIAndReconcile()</b> method from time to time, following the re-importation of controlled packages - for example, after using <b>Package.VersionControlGetLatest()</b> to update a number of packages, or after performing a number of package check-outs</p>                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>VersionControlGetLatest</b><br>(boolean ForceImport) | Void | <p>Updates the local working copy of the package file associated with the object package, before re-importing the package data from the package file</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ForceImport: Boolean - Used if the package data in the model is found to be up-to-date with respect to the version controlled package file; if: <ul style="list-style-type: none"> <li>• <b>False</b>, the package data that exists in the model is accepted as being up-to-date and no attempt is made to re-import data from the package file</li> <li>• <b>True</b>, Enterprise Architect re-imports the package from the package file regardless</li> </ul> </li> </ul> <p>See also the version control menu option <b>Get Latest</b></p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke the <b>Repository.ScanXMIAndReconcile()</b> method from time to time, following the re-importation of controlled packages - for example, after using <b>Package.VersionControlGetLatest()</b> to update a number of packages, or after performing a number of package check-outs</p> |
| <b>VersionControlGetStatus ()</b>                       | Long | <p>Returns the version control status of the package. Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Method                                                        | Type        | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                               |             | <p>Return value maps to the following enumerated type:</p> <pre>enum EnumCheckOutStatus {     csUncontrolled = 0,     csCheckedIn,     csCheckedOutToThisUser,     csReadOnlyVersion,     csCheckedOutToAnotherUser,     csOfflineCheckedIn,     csCheckedOutOfflineByUser,     csCheckedOutOfflineByOther,     csDeleted, };</pre> <p><i>csUncontrolled</i> - Either unable to communicate with the version control provider associated with the package or the package file is unknown to the provider.</p> <p><i>csReadOnlyVersion</i> - Package is marked as read-only. An earlier revision of the package has been retrieved from version control.</p> <p><i>csOfflineCheckedOutToThisUser</i> - Indicates that the package was 'checked out' by this user whilst disconnected from version control.</p> <p><i>csOfflineNotCheckedOutToThisUser</i> - Indicates that Enterprise Architect can not currently connect to the version control config and the package was not previously checked out to this user.</p> <p><i>csDeleted</i> - The package file has been deleted from version control.</p> |
| <b>VersionControlPutLatest (string CheckInComment)</b>        | <i>Void</i> | <p>Perform a checkin of the version controlled package, whilst keeping the package checked-out.</p> <p>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.</p> <p>When a package that was previously marked as <i>Checked Out Offline</i>, is successfully 'Put' (checkedin) to version control, that package's flags are updated to clear the <i>Checked Out Offline</i> indicator.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message added to the version controlled file's history (where applicable).</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>VersionControlRemove ()</b>                                | <i>Void</i> | <p>Removes version control from the package.</p> <p>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>VersionControlResynchPkgStatus (boolean ClearSettings)</b> |             | <p><b>Synchronizes</b> the version control status of the single object package recorded in your current model with the package status reported by your version control provider.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ClearSettings: Boolean - used if the package file associated with the specified package is reported by the</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Method | Type | Notes                                                                                                                                                                                                                                                            |
|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |      | version control provider as uncontrolled. If ClearSettings is: <ul style="list-style-type: none"> <li>• <b>True</b>, the version control settings are cleared from the package</li> <li>• <b>False</b>, the version control settings remain unchanged</li> </ul> |

#### Learn More:

- [Repository Models](#) <sup>[1856]</sup>
- [Package Packages](#) <sup>[1847]</sup>
- [Version Control - Get Latest](#) <sup>[279]</sup>
- [Resynchronize the Status of Version Controlled Packages](#) <sup>[297]</sup>
- [Preserving Cross Package References](#) <sup>[335]</sup>

#### 18.2.2.4.9 *ProjectIssues Class*

A system-level Issue. Indicates a problem or risk associated with the system as a whole. Accessed using the Repository *Issues* collection.

Associated table in .EAP file: *t\_issues*

#### ProjectIssues Attributes

| Attribute           | Type                                         | Notes                                                                     |
|---------------------|----------------------------------------------|---------------------------------------------------------------------------|
| <b>Category</b>     | <i>String</i>                                | Read/Write. The category this issue belongs to.                           |
| <b>Date</b>         | <i>Date</i>                                  | Read/Write. Date created.                                                 |
| <b>DateResolved</b> | <i>Date</i>                                  | Read/Write. Date issue resolved.                                          |
| <b>Name</b>         | <i>String</i>                                | Read/Write. Issue name (that is, the issue itself).                       |
| <b>IssueID</b>      | <i>Long</i>                                  | Read only. The ID of this issue.                                          |
| <b>Notes</b>        | <i>String</i>                                | Read/Write. Associated description of issue.                              |
| <b>ObjectType</b>   | <a href="#">ObjectType</a> <sup>[1835]</sup> | Read only. Distinguishes objects referenced through a Dispatch interface. |
| <b>Owner</b>        | <i>String</i>                                | Read/Write. Owner of issue.                                               |
| <b>Priority</b>     | <i>String</i>                                | Read/Write. Issue priority. Generally should use Low, Medium or High.     |
| <b>Resolution</b>   | <i>String</i>                                | Read/Write. Description of resolution.                                    |
| <b>Resolver</b>     | <i>String</i>                                | Read/Write. Person resolving issue.                                       |
| <b>Severity</b>     | <i>String</i>                                | Read/Write. Issue severity. Should be marked as Low, Medium or High.      |
| <b>Status</b>       | <i>String</i>                                | Read/Write. Current issue status.                                         |

#### ProjectIssues Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Issue object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                 |

#### 18.2.2.4.10 ProjectResource Class

A *Project Resource* is a named person who is available to work on the current project in any capacity. Accessed using the Repository *Resources* collection.

Associated table in .EAP file: *t\_resources*

#### ProjectResource Attributes

| Attribute           | Type                                                         | Notes                                                                     |
|---------------------|--------------------------------------------------------------|---------------------------------------------------------------------------|
| <b>Email</b>        | <i>String</i>                                                | Email address.                                                            |
| <b>Fax</b>          | <i>String</i>                                                | Fax number.                                                               |
| <b>Mobile</b>       | <i>Variant</i>                                               | Mobile number if available.                                               |
| <b>Name</b>         | <i>String</i>                                                | Name of resource.                                                         |
| <b>Notes</b>        | <i>String</i>                                                | A description if appropriate.                                             |
| <b>ObjectType</b>   | <a href="#">ObjectType</a> <sup>[1835]</sup>                 | Read only. Distinguishes objects referenced through a Dispatch interface. |
| <b>Organization</b> | <a href="#">Package</a> <sup>[1845]</sup> :<br><i>String</i> | Organization resource associated with.                                    |
| <b>Phone1</b>       | <i>Variant</i>                                               | Main phone.                                                               |
| <b>Phone2</b>       | <i>Variant</i>                                               | Alternative phone.                                                        |
| <b>Roles</b>        | <i>String</i>                                                | The roles this resource can play in the current project.                  |

#### ProjectResource Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Resource object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more                           |

| Method | Type | Notes        |
|--------|------|--------------|
|        |      | information. |

#### 18.2.2.4.11 ProjectRole Class

A *ProjectRole* object represents a named project role. Accessed using the Repository *ProjectRole* collection.

Associated table in .EAP file: *t\_projectroles*

##### ProjectRole Attributes

| Attribute   | Type                                 | Notes                                                                     |
|-------------|--------------------------------------|---------------------------------------------------------------------------|
| Description | String                               | Read/Write. The project role item description.                            |
| Notes       | String                               | Read/Write. Notes about the project role item.                            |
| ObjectType  | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface. |
| Role        | String                               | Read/Write. The project role item name.                                   |

##### ProjectRole Methods

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Updates the current ProjectRole object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.          |

#### 18.2.2.4.12 PropertyType Class

A *PropertyType* object represents a defined property that can be applied to UML elements as a Tagged Value. Accessed using the Repository *PropertyTypes* collection. Each *PropertyType* corresponds to one of the predefined Tagged Values for the model.

Associated table in .EAP file: *t\_propertytypes*

##### Author Attributes

| Attribute   | Type                                 | Notes                                                          |
|-------------|--------------------------------------|----------------------------------------------------------------|
| Description | String                               | Read/Write. Short description for the property.                |
| Detail      | String                               | Read/Write. Configuration information for the property.        |
| ObjectType  | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch |

| Attribute  | Type          | Notes                                        |
|------------|---------------|----------------------------------------------|
|            |               | interface.                                   |
| <b>Tag</b> | <i>String</i> | Read/Write. Name of the property (Tag Name). |

#### Author Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>PropertyType</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.   |

#### 18.2.2.4.13 Reference Class

This Interface provides access to the various lookup tables within Enterprise Architect. Use the Repository *GetReferenceList()* method to get a handle to a list. Valid lists are:

- Diagram
- Element
- Constraint
- Requirement
- Connector
- Status
- Cardinality
- Effort
- Metric
- Scenario
- Status
- Test

#### Reference Attributes

| Attribute         | Type                                                | Notes                                                                     |
|-------------------|-----------------------------------------------------|---------------------------------------------------------------------------|
| <b>Count</b>      | <i>Short</i>                                        | Count of items in the list.                                               |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                                       | The list type (for example, DiagramTypes).                                |

#### Reference Methods

| Method                     | Type          | Notes                                                                                                                                                                                 |
|----------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetAt (short Index)</b> | <i>String</i> | Get the item at the specified index.<br>Parameters:<br><ul style="list-style-type: none"> <li>Index: Short - The index of the item to retrieve from the list.</li> </ul>              |
| <b>GetLastError ()</b>     | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Refresh ()</b>          | <i>Short</i>  | Refresh the current list and return the count of items.                                                                                                                               |

#### 18.2.2.4.14 Repository Class

The *Repository* is the main container of all structures such as models, packages and elements. You can iteratively begin accessing the model using the *Models* collection. It also has some convenient methods to directly access the structures without having to locate them in the hierarchy first.

Associated table in .EAP file: <none>

#### Repository Attributes:

| Attribute               | Type                                                     | Notes                                                                                                                                                                                                                                                                                                                        | See Also |
|-------------------------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Authors</b>          | <a href="#">Collection</a><br><small>[1841]</small>      | Read only. The system <i>Authors</i> collection. Contains 0 or more <i>Author objects</i> , each of which can be associated with, for example, elements or diagrams as the item author or owner. Use <i>AddNew</i> , <i>Delete</i> and <i>GetAt</i> to manage Authors.                                                       |          |
| <b>BatchAppend</b>      | <i>Boolean</i>                                           | Read/Write. Set this property to <b>true</b> when your automation client has to rapidly insert many elements, operations, attributes and/or operation parameters. Set to <b>false</b> when work is complete.<br><br>This can result in 10- to 20-fold improvement in adding new elements in bulk.                            |          |
| <b>Clients</b>          | <a href="#">Collection</a><br><small>[1841]</small>      | Read only. A list of <i>Clients</i> associated with the project. You can modify, delete and add new <i>Client objects</i> using this collection.                                                                                                                                                                             |          |
| <b>ConnectionString</b> | <i>String</i>                                            | Read only. The filename/connection string of the current Repository.                                                                                                                                                                                                                                                         |          |
| <b>Datatypes</b>        | <a href="#">Collection</a><br><small>[1841]</small>      | Read only. The <i>Datatypes</i> collection. Contains a list of <i>Datatype objects</i> , each representing a data type definition for either data modeling or code generation purposes.                                                                                                                                      |          |
| <b>EAEdition</b>        | <a href="#">EAEdition Types</a><br><small>[1833]</small> | Read only. Returns the current level of core licensed functionality available<br><br>This property returns <b>Corporate</b> when the edition is <i>Business and Software Engineering</i> , <i>Systems Engineering</i> or <i>Ultimate</i> . Use <b>EAEditionEx</b> to identify which of these extended editions is available. |          |

| Attribute         | Type                                                                                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also                                              |
|-------------------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| EAEditionEx       | <a href="#">EAEditionTypes</a> <sup>[1833]</sup>                                                     | Read only. Returns the current level of extended licensed functionality available                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                       |
| EnableCache       | Boolean                                                                                              | Read/Write. An optimization for pre-loading package objects when dealing with large sets of automation objects.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                       |
| EnableUIUpdates   | Boolean                                                                                              | Read/Write. Set this property to <b>false</b> to improve the performance of changes to the model; for example, bulk addition of elements to a package. To reveal the changes to the user, call <i>Repository.RefreshModelView()</i> .                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                       |
| FlagUpdate        | Boolean                                                                                              | Read/Write. Instructs Enterprise Architect to update the Repository with the <i>LastUpdate</i> value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                       |
| InstanceGUID      | String                                                                                               | Read only. The identifier string identifying the Enterprise Architect runtime session.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                       |
| IsSecurityEnabled | Boolean                                                                                              | Read only. Checks whether User Security is enabled for the current repository.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                       |
| Issues            | <a href="#">Collection</a> <sup>[1841]</sup>                                                         | Read only. The <i>System Issues</i> list. Contains <i>ProjectIssues</i> objects, each detailing a particular issue as it relates to the project as a whole.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                       |
| LastUpdate        | String                                                                                               | Read only. The identifier string identifying the Enterprise Architect runtime session and the timestamp for when it was set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                       |
| LibraryVersion    | Long                                                                                                 | Read only. The build number of the Enterprise Architect runtime.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                       |
| Models            | <a href="#">Collection</a> <sup>[1841]</sup><br>of type<br><a href="#">Package</a> <sup>[1845]</sup> | <p>Read only. <i>Models</i> are of type <i>package</i> and belong to a collection of packages. This is the top level entry point to an Enterprise Architect project file. Each model is a <i>root node</i> in the <b>Project Browser</b> and can contain items such as Views and packages.</p> <p>A model is a special form of a package; it has a <i>ParentID</i> of <b>0</b>. By iterating through all models, you can access all the elements within the project hierarchy.</p> <p>You can also use the <i>AddNew</i> function to create a new model. A model can be deleted, but remember that everything contained in the model is deleted as well.</p> |                                                       |
| ObjectType        | <a href="#">ObjectType</a> <sup>[1835]</sup>                                                         | Read only. Distinguishes objects referenced through the Dispatch interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                       |
| ProjectGUID       | String                                                                                               | Read only. Returns a unique ID for the project.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                       |
| ProjectRoles      | <a href="#">Collection</a> <sup>[1841]</sup>                                                         | Read only. The system Roles collection. Contains <b>0</b> or more Role objects, each of which can be associated with, for example, elements or diagrams as the item author or owner. Use <i>AddNew</i> , <i>Delete</i> and <i>GetAt</i> to manage Roles                                                                                                                                                                                                                                                                                                                                                                                                      |                                                       |
| PropertyTypes     | <a href="#">Collection</a> <sup>[1841]</sup>                                                         | Read only. Collection of <b>Property Types</b> available to the Repository.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <a href="#">Property Type Class</a> <sup>[1853]</sup> |

| Attribute         | Type                                                | Notes                                                                                                                                                                                                                                                                       | See Also                                                                                                                       |
|-------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| Resources         | <a href="#">Collection</a><br><small>[1841]</small> | Read only. Contains available <i>ProjectResource</i> objects to assign to work items within the project. Use the <b>add new</b> , <b>modify</b> and <b>delete</b> functions to manage resources.                                                                            |                                                                                                                                |
| Stereotypes       | <a href="#">Collection</a><br><small>[1841]</small> | Read only. The <b>Stereotype</b> collection. A list of <i>Stereotype objects</i> that contain information on a stereotype and which elements it can be applied to.                                                                                                          | <a href="#">Stereotype Class</a><br><small>[1871]</small>                                                                      |
| SuppressEADialogs | Boolean                                             | Read/Write. Set this property in the <b>EA_OnPostNewElement</b> or <b>EA_OnPostNewConnector</b> broadcast events to control whether Enterprise Architect should suppress showing the default <b>Properties</b> dialogs to the user when an element or connector is created. | <a href="#">EA_OnPostNewElement</a><br><small>[2011]</small><br><a href="#">EA_OnPostNewConnector</a><br><small>[2011]</small> |
| Tasks             | <a href="#">Collection</a><br><small>[1841]</small> | Read only. A list of system tasks (to do list). Each entry is a <b>Task</b> Item; you can modify, delete and add new tasks.                                                                                                                                                 | <a href="#">Task Class</a><br><small>[1872]</small>                                                                            |
| Terms             | <a href="#">Collection</a><br><small>[1841]</small> | Read only. The project <i>Glossary</i> . Each <b>Term</b> object is an entry in the Glossary. Add, modify and delete Terms to maintain the Glossary.                                                                                                                        | <a href="#">Term Class</a><br><small>[1873]</small>                                                                            |

**Repository Methods:**

| Method                                         | Type    | Notes                                                                                                                                                                                                                                                                                     | See Also |
|------------------------------------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| ActivateDiagram (Long DiagramID)               |         | Activates an already open diagram (that is, makes it the active tab) in the main Enterprise Architect user interface.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to make active.</li> </ul>                                     |          |
| ActivatePerspective (string, long)             | Boolean | <b>Deprecated</b> - no longer in use.                                                                                                                                                                                                                                                     |          |
| ActivateTab (string Name)                      |         | Activates an open Enterprise Architect tabbed view.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>Name: String - the name of the view to activate.</li> </ul>                                                                                                              |          |
| ActivateTechnology (string Name)               |         | Activates an enabled MDG Technology<br><br>Parameters:<br><ul style="list-style-type: none"> <li>Name: String - the name of the Technology to activate</li> </ul>                                                                                                                         |          |
| ActivateToolbox (string Toolbox, long Options) | Boolean | Activates a Toolbox page in the GUI<br><br>Returns <b>true</b> if the specified Toolbox page is successfully activated, otherwise returns <b>false</b><br><br>Parameters:<br><ul style="list-style-type: none"> <li>Toolbox: String - the name of the Toolbox page to activate</li> </ul> |          |

| Method                                                   | Type                          | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also                                         |
|----------------------------------------------------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
|                                                          |                               | <ul style="list-style-type: none"> <li>Options: Long - reserved for future use</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                  |
| <b>AddDefinedSearches (string sXML)</b>                  |                               | <p>Enables you to enter a set of defined searches that last in Enterprise Architect for the life of the application; when Enterprise Architect loads again they must be inserted again by your Add-In</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>sXML: String - the XML of the defined searches; you can get this XML by performing an <i>export</i> of the searches from the Manage Searches dialog in Enterprise Architect</li> </ul>                                                                                                                                                               | <a href="#">Create and Manage Searches</a> [482] |
| <b>AddPerspective (string Perspective, long Options)</b> | <i>Boolean</i>                | <b>Deprecated</b> - no longer in use                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                  |
| <b>AddTab (string TabName, string ControlID)</b>         | <i>activeX custom control</i> | <p>Adds an ActiveX custom control as a tabbed window. Enterprise Architect creates a control and, if successful, returns its Unknown pointer, which can be used by the caller to manipulate the control.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>TabName: String - used as the tab caption</li> <li>ControlID: String - the ProgID of the control; for example, Project1,UserControl1</li> </ul>                                                                                                                                                                                                   |                                                  |
| <b>AddWindow (string WindowName, string ControlID)</b>   | <i>activeX custom control</i> | <p>Adds an ActiveX custom control as a window to the Add-Ins docked window. Enterprise Architect creates a control and, if successful, returns its Unknown pointer, which can be used by the caller to manipulate the control.</p> <p>The window can be shown by selecting it from the list in the Workspace Layouts toolbar - click on the third icon from the right and look at the end of the list.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>WindowName: String - used as the window title</li> <li>ControlID: String - the ProgID of the control; for example, Project1,UserControl1</li> </ul> | <a href="#">Workspace Layouts</a> [115]          |
| <b>AdviseConnectorChange (long ConnectorID)</b>          |                               | <p>Provides an Add-In or automation client with the ability to advise the Enterprise Architect user interface that a particular connector has changed and, if it is visible in any open diagram, to reload and refresh that connector for the user.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ConnectorID: Long - the ID of the connector.</li> </ul>                                                                                                                                                                                                                                                |                                                  |
| <b>AdviseElementChange (long ObjectID)</b>               |                               | <p>Provides an Add-In or automation client with the ability to advise the Enterprise Architect user interface that a particular element has changed and, if it is visible in any open diagram, to reload and refresh that element for the user.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ObjectID: Long - the ID of the element.</li> </ul>                                                                                                                                                                                                                                                         |                                                  |

| Method                                                                                       | Type                                        | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See Also                                                                                                                            |
|----------------------------------------------------------------------------------------------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>ChangeLoginUser</b><br>(string Name,<br>string Password)                                  | <i>Boolean</i>                              | Sets the currently logged on user to be that specified by a name and password; this logs the user into the repository when security is enabled<br><br>If security is not enabled an exception ( <i>Security not enabled</i> ) is thrown<br><br>Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the user</li> <li>Password: String - the password of the user</li> </ul>                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                     |
| <b>ClearAuditLogs</b><br>(Object<br>StartDateTime,<br>Object<br>EndDateTime)                 | <i>Boolean</i>                              | Clears all Audit Logs from the model<br><br>If <i>StartDateTime</i> and <i>EndDateTime</i> are not null then only log items that fall into this period are cleared<br><br>Returns <b>true</b> for success, <b>false</b> for failure <ul style="list-style-type: none"> <li>This method cannot be undone. It is strongly advised that you call <i>SaveAuditLogs</i> first to backup the logs</li> <li>This method might fail if the user logged into the model does not have the correct access permission</li> </ul> Parameters: <ul style="list-style-type: none"> <li>StartDateTime: Variant ( DateTime ) - the earliest date and time of log entries to clear.</li> <li>EndDateTime; Variant ( DateTime ) - the latest date and time of log entries to clear.</li> </ul> |                                                                                                                                     |
| <b>ClearOutput</b> (string<br>Name)                                                          |                                             | Removes all the text from a tab in the Output window<br><br>Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the tab to remove text from</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <a href="#">CreateOutputTab</a><br>[1860]<br><a href="#">EnsureOutputVisible</a><br>[1860]<br><a href="#">WriteOutput</a><br>[1871] |
| <b>CloseAddins</b> ()                                                                        |                                             | Called by automation controllers to ensure that Add-Ins created in .NET do not linger after all controller references to Enterprise Architect have been cleared                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                     |
| <b>CloseDiagram</b><br>(long DiagramID)                                                      |                                             | Closes a diagram in the current list of diagrams that Enterprise Architect has open<br><br>Parameters: <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to close</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                     |
| <b>CloseFile</b> ()                                                                          |                                             | Closes any open file                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                     |
| <b>CreateDocumentGenerator</b> ()                                                            | <a href="#">DocumentGenerator</a><br>[1947] | Returns a pointer to the <i>EA.DocumentGenerator</i> interface                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                     |
| <b>CreateModel</b><br>(CreateModelType<br>CreateType, string<br>FilePath, long<br>ParentWnd) | <i>Boolean</i>                              | Creates a new .eap model file based on the standard Enterprise Architect Base model, or a shortcut .eap based on a provided SQL connection<br><br>Returns <b>true</b> when the new file is created, otherwise                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <a href="#">CreateModelType</a><br>[1832]                                                                                           |

| Method                                                                       | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also                                                                                                                            |
|------------------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
|                                                                              |                | <p>returns <b>false</b></p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• CreateType: CreateModelType - Specify whether to make a new copy of the EABase.eap model, or create a .eap file shortcut to a DBMS repository. The latter option requires a dialog to be opened for the user to provide SQL connection details.</li> <li>• FilePath: String - Destination for new .eap file</li> <li>• ParentWnd: Long - Window handle to act as the parent for the SQL connection dialog; only required when using <i>cmEAPFromSQLRepository</i>.</li> </ul>  |                                                                                                                                     |
| <b>CreateOutputTab</b><br>(string Name)                                      |                | <p>Creates a tab in the <b>Output</b> window.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Name: String - the name of the tab to create.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                 | <a href="#">CreateOutputTab</a><br>[1860]<br><a href="#">EnsureOutputVisible</a><br>[1860]<br><a href="#">WriteOutput</a><br>[1871] |
| <b>DeletePerspective</b><br>(string Perspective, long Options)               | <i>Boolean</i> | <b>Deprecated</b> - no longer in use                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                     |
| <b>DeleteTechnology</b><br>(string ID)                                       | <i>Boolean</i> | <p>Removes a specified MDG Technology resource from the repository.</p> <p>Returns <b>true</b>, if the technology is successfully removed from the model. Returns <b>false</b> otherwise.</p> <ul style="list-style-type: none"> <li>• This applies to technologies imported into pre-7.0 versions of Enterprise Architect (imported technologies), not to technologies referenced in version 7.0 and later (referenced technologies).</li> </ul> <p><b>Parameters:</b></p> <ul style="list-style-type: none"> <li>• ID: String - the ID of the technology.</li> </ul> | <a href="#">Deploy an MDG Technology</a><br>[1089]                                                                                  |
| <b>EnsureOutputVisible</b><br>(string Name)                                  |                | <p>Ensures that a specified tab in the <b>Output</b> window is visible to the user. The <b>Output</b> window is made visible if it is hidden.</p> <p><b>Parameters:</b></p> <ul style="list-style-type: none"> <li>• Name: String - the name of the tab to make visible.</li> </ul>                                                                                                                                                                                                                                                                                    | <a href="#">CreateOutputTab</a><br>[1860]<br><a href="#">EnsureOutputVisible</a><br>[1860]<br><a href="#">WriteOutput</a><br>[1871] |
| <b>ExecutePackageBuildScript</b><br>(long ScriptOptions, string PackageGuid) |                | <p>Enables you to run the active package build script based on your current selection in the <b>Project Browser</b>. You can also run a script by passing in the package GUID.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ScriptOptions: Long - the script type; can be any one of these numerical values:<br/>           1 = Build<br/>           2 = Test</li> </ul>                                                                                                                                                                            |                                                                                                                                     |

| Method                              | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                    | See Also |
|-------------------------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                     |                                                     | <p>3 = Run<br/>4 = Create Workbench Instance<br/>5 = Debug.</p> <ul style="list-style-type: none"> <li>PackageGuid: String - the ID of the package for which to run the script.</li> </ul>                                                                                                                                                                                               |          |
| Exit                                |                                                     | Shuts down Enterprise Architect immediately. Used by .NET programmers where the garbage collector does not immediately release all referenced COM objects.                                                                                                                                                                                                                               |          |
| GetActivePerspective ()             | String                                              | <b>Deprecated</b> - no longer in use.                                                                                                                                                                                                                                                                                                                                                    |          |
| GetAttributeByGuid (string Guid)    | <a href="#">Attribute</a><br><small>[1898]</small>  | Returns a pointer to an attribute in the repository, located by its GUID. This is usually found using the <i>AttributeGUID</i> property of an attribute.<br><br>Parameters: <ul style="list-style-type: none"> <li>Guid: String - the GUID of the attribute to locate.</li> </ul>                                                                                                        |          |
| GetAttributeById (string Id)        | <a href="#">Attribute</a><br><small>[1898]</small>  | Returns a pointer to an attribute in the repository, located by its ID. This is usually found using the <i>AttributeID</i> property of an attribute.<br><br>Parameters: <ul style="list-style-type: none"> <li>Id: String - the ID of the attribute to locate.</li> </ul>                                                                                                                |          |
| GetConnectorByGuid (string Guid)    | <a href="#">Connector</a><br><small>[1911]</small>  | Returns a pointer to a connector in the repository, located by its GUID. This is usually found using the <i>ConnectorGUID</i> property of a connector.<br><br>Parameters: <ul style="list-style-type: none"> <li>Guid: String - the GUID of the connector to locate.</li> </ul>                                                                                                          |          |
| GetConnectorById (long ConnectorID) | <a href="#">Connector</a><br><small>[1911]</small>  | Searches the repository for a connector with a specific ID.<br><br>Parameters: <ul style="list-style-type: none"> <li>ConnectorID: Long - the ID of the connector to locate.</li> </ul>                                                                                                                                                                                                  |          |
| GetContextItem (object Item)        | <a href="#">ObjectType</a><br><small>[1835]</small> | Sets a pointer to an item in context within Enterprise Architect.<br><br>Also returns the corresponding <i>ObjectType</i> .<br><br>For additional information about <i>ContextItems</i> and the supported <i>ObjectTypes</i> see the <b>GetContextItemType</b> method (below).<br><br>Parameters: <ul style="list-style-type: none"> <li>Item: Object - the item to point to.</li> </ul> |          |
| GetContextItemType ()               | <a href="#">ObjectType</a><br><small>[1835]</small> | Returns the <i>ObjectType</i> of an item in context within Enterprise Architect. A <i>ContextItem</i> is defined as an item selected anywhere within the Enterprise Architect GUI including: <ul style="list-style-type: none"> <li>An item selected in the <b>Project Browser</b></li> </ul>                                                                                            |          |

| Method                                               | Type                                             | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See Also |
|------------------------------------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                      |                                                  | <ul style="list-style-type: none"> <li>An item selected in an open diagram</li> <li>An item selected in certain dialogs, such as the attribute <b>Properties</b> dialog.</li> </ul> <p>The supported <i>ObjectTypes</i> can be any one of the following values:</p> <ul style="list-style-type: none"> <li><i>otElement</i></li> <li><i>otPackage</i></li> <li><i>otDiagram</i></li> <li><i>otAttribute</i></li> <li><i>otMethod</i></li> <li><i>otConnector</i></li> </ul> |          |
| <b>GetCurrentObject ()</b>                           | <i>Object</i>                                    | Returns the current context Object.                                                                                                                                                                                                                                                                                                                                                                                                                                         |          |
| <b>GetCounts ()</b>                                  | <i>String</i>                                    | Returns a set of counts from a number of tables within the base Enterprise Architect repository. These can be used to determine whether records have been added or deleted from the tables for which information is retrieved.                                                                                                                                                                                                                                              |          |
| <b>GetCurrentDiagram ()</b>                          | <a href="#">Diagram</a><br><small>[1927]</small> | Returns a selected diagram.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |          |
| <b>GetCurrentLoginUser (boolean GetGuid = false)</b> | <i>String</i>                                    | If security is not enabled in the repository, an error is generated.<br><br>If <i>GetGuid</i> is <b>True</b> , a GUID generated by Enterprise Architect representing the user is returned; otherwise the text as entered in <i>System Users/User Details/Login</i> is returned.                                                                                                                                                                                             |          |
| <b>GetDiagramByGuid (string Guid)</b>                | <a href="#">Diagram</a><br><small>[1927]</small> | Returns a pointer to a diagram using the global reference ID (global ID). This is usually found using the diagram <i>GUID</i> property of an element, and stored for later use to open a diagram without using the collection <i>GetAt()</i> function.<br><br>Parameters: <ul style="list-style-type: none"> <li>Guid: String - the GUID of the diagram to locate.</li> </ul>                                                                                               |          |
| <b>GetDiagramByID (long DiagramID)</b>               | <a href="#">Diagram</a><br><small>[1927]</small> | Gets a pointer to a diagram using an absolute reference number (local ID). This is usually found using the <i>DiagramID</i> property of an element, and stored for later use to open a diagram without using the collection <i>GetAt()</i> function.<br><br>Parameters: <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to locate.</li> </ul>                                                                                                |          |
| <b>GetElementByGuid (string Guid)</b>                | <a href="#">Element</a><br><small>[1877]</small> | Returns a pointer to an element in the repository, using the element's GUID reference number (global ID). This is usually found using the <i>ElementGUID</i> property of an element, and stored for later use to open an element without using the collection <i>GetAt()</i> function.<br><br>Parameters:                                                                                                                                                                   |          |

| Method                                                          | Type                                               | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also |
|-----------------------------------------------------------------|----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                 |                                                    | <ul style="list-style-type: none"> <li>Guid: String - the GUID of the element to locate.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |          |
| <b>GetElementByID (long ElementID)</b>                          | <a href="#">Element</a><br><small>[187]</small>    | <p>Gets a pointer to an element using an absolute reference number (local ID). This is usually found using the <i>ElementID</i> property of an element, and stored for later use to open an element without using the collection <i>GetAt()</i> function.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementID: Long - the ID of the element to locate.</li> </ul>                                                                                                                                                                                                                                                  |          |
| <b>GetElementsByQuery (string QueryName, string SearchTerm)</b> |                                                    | <p>Enables the user to run a search in Enterprise Architect, returning the result as a collection.</p> <p>For example <i>GetElementsByQuery ('Simple','Class1')</i>, where results contain elements with <i>Class1</i> in the <b>Name</b> and <b>Notes</b> fields.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>QueryName: String - the name of the search to run, for example 'Simple'.</li> <li>SearchTerm: String - the term to search for.</li> </ul>                                                                                                                                                             |          |
| <b>GetElementSet (string IDList, long Options)</b>              | <a href="#">Collection</a><br><small>[184]</small> | <p>Returns a set of elements as a collection based on a comma-separated list of <i>ElementID</i> values. By default, if no values are provided in the IDList parameter, all objects for the entire project are returned.</p> <p>Parameters</p> <ul style="list-style-type: none"> <li>IDList: String - a comma-separated list of <i>ElementID</i> values</li> <li>Options: Long - modifies default behaviour of this method <ul style="list-style-type: none"> <li>1 - Returns empty collection when empty IDList parameter is given</li> <li>2 - Use IDList string as an SQL query to populate this collection</li> </ul> </li> </ul> |          |
| <b>GetFieldFromFormat (string Format, string Text)</b>          | String                                             | <p>Converts a field from your preferred format to Enterprise Architect's internal format; returns the field in Enterprise Architect's internal format</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Format: String - The format to convert the field from; valid formats are: <ul style="list-style-type: none"> <li>HTML - Full HTML</li> <li>RTF - Rich Text Format</li> <li>TXT - Plain text</li> </ul> </li> <li>Text: String - The field to be converted</li> </ul>                                                                                                                                               |          |
| <b>GetFormatFromField (string Format, string Text)</b>          | String                                             | <p>After accessing a field that contains formatting, use this method to convert it to your preferred format; returns the field in the format specified</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Format: String - The format to convert the field to; valid formats are:</li> </ul>                                                                                                                                                                                                                                                                                                                                |          |

| Method                                 | Type                                      | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See Also                                                |
|----------------------------------------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
|                                        |                                           | <ul style="list-style-type: none"> <li>HTML - Full HTML</li> <li>RTF - Rich Text Format</li> <li>TXT - Plain text</li> <li>Text: String - The field to be converted</li> </ul>                                                                                                                                                                                                                                                                                                                                                                            |                                                         |
| <b>GetLastError ()</b>                 | <i>String</i>                             | <p>Returns a string value describing the most recent error that occurred in relation to this object</p> <p>This function is rarely used as an exception is thrown when an error occurs</p>                                                                                                                                                                                                                                                                                                                                                                |                                                         |
| <b>GetMailInterface()</b>              | <i>MailInterface</i>                      | Returns an instance of the <i>EA.MailInterface</i> ; use this interface to automate the process of creating and sending model mail messages                                                                                                                                                                                                                                                                                                                                                                                                               | <a href="#">MailInterface Package</a> <sup>[1950]</sup> |
| <b>GetMethodByGuid (string Guid)</b>   | <a href="#">Method</a> <sup>[1903]</sup>  | <p>Returns a pointer to a method in the repository; this is usually found using the <i>MethodGUID</i> property of a method</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Guid: String - the GUID of the method to look for</li> </ul>                                                                                                                                                                                                                                                                                                     |                                                         |
| <b>GetMethodById (string Id)</b>       | <a href="#">Method</a> <sup>[1903]</sup>  | <p>Returns a pointer to a method in the repository; this is usually found using the <i>MethodID</i> property of a method</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Id: String - the ID of the method to look for</li> </ul>                                                                                                                                                                                                                                                                                                           |                                                         |
| <b>GetPackageByGuid (string Guid)</b>  | <a href="#">Package</a> <sup>[1845]</sup> | <p>Returns a pointer to a package in the repository using the package's GUID reference number (global ID). This is usually found using the <i>PackageGUID</i> property of the package.</p> <p>Each package in the model also has an associated element with the same GUID, so if you have an element with <i>Type="Package"</i> then you can load the package by calling:</p> <p><i>GetPackageByGuid(Element.ElementGUID)</i></p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Guid: String - the GUID of the package to look for</li> </ul> |                                                         |
| <b>GetPackageById (long PackageID)</b> | <a href="#">Package</a> <sup>[1845]</sup> | <p>Get a pointer to a package using an absolute reference number (local ID). This is usually found using the <i>PackageID</i> property of an package, and stored for later use to open a package without using the collection <i>GetAt()</i> function.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageID: Long - the ID of the package to locate</li> </ul>                                                                                                                                                                         |                                                         |
| <b>GetProjectInterface ()</b>          | <a href="#">Project</a> <sup>[1929]</sup> | Return a pointer to the <i>EA.Project interface</i> (the XML-based automation server for Enterprise Architect). Use this interface to work with Enterprise Architect using XML, and also to access utility functions for loading diagrams, running reports and                                                                                                                                                                                                                                                                                            | <a href="#">EA.Project interface</a> <sup>[1929]</sup>  |

| Method                                           | Type                                         | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See Also                                              |
|--------------------------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
|                                                  |                                              | so on.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                       |
| <b>GetReferenceList (string Type)</b>            | <a href="#">Reference</a> <sup>[1859]</sup>  | Uses the list type to get a pointer to a <i>Reference List</i> object.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>Type: String - specifies the list type to get; valid list types are:<br/><br/><i>Diagram</i><br/><i>Element</i><br/><i>Constraint</i><br/><i>Requirement</i><br/><i>Connector</i><br/><i>Status</i><br/><i>Cardinality</i><br/><i>Effort</i><br/><i>Metric</i><br/><i>Scenario</i><br/><i>Status</i> and<br/><i>Test</i>.</li> </ul>                                                        |                                                       |
| <b>GetTechnologyVersion (string ID)</b>          | <i>String</i>                                | Returns the version of a specified MDG Technology resource.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>ID: String - the specified technology ID.</li> </ul>                                                                                                                                                                                                                                                                                                                                                   |                                                       |
| <b>GetTreeSelectedElements()</b>                 | <a href="#">Collection</a> <sup>[1847]</sup> | Returns the set of elements currently selected in the <b>Project Browser</b> as a collection.                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                       |
| <b>GetTreeSelectedItem (object SelectedItem)</b> | <a href="#">ObjectType</a> <sup>[1835]</sup> | Gets an object variable and type corresponding to the currently selected item in the tree view.<br><br>To use this function, create a generic object variable and pass this as the parameter. Depending on the return type, cast it to a more specific type.<br><br>The object passed back through the parameter can be a package, element, diagram, attribute or operation object.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>SelectedItem: Object - the object to get the variable and type for.</li> </ul> |                                                       |
| <b>GetTreeSelectedItemType ()</b>                | <a href="#">ObjectType</a> <sup>[1835]</sup> | Returns the type of the object currently selected in the tree. One of:<br><ul style="list-style-type: none"> <li><i>otDiagram</i></li> <li><i>otElement</i></li> <li><i>otPackage</i></li> <li><i>otAttribute</i></li> <li><i>otMethod</i>.</li> </ul>                                                                                                                                                                                                                                                                          |                                                       |
| <b>GetTreeSelectedObject ()</b>                  | <i>Object</i>                                | The related method <b>GetTreeSelectedItem()</b> has an output parameter that is inaccessible by some scripting languages. As an alternative, this method provides the selected item through the return value.                                                                                                                                                                                                                                                                                                                   | <a href="#">GetTreeSelectedItem</a> <sup>[1865]</sup> |
| <b>GetTreeSelectedPackage</b>                    | <a href="#">Package</a>                      | Returns the package in which the currently selected                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                       |

| Method                                                                | Type                                                                              | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See Also                                                                                                                     |
|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| ckage ()                                                              |  | tree view object is contained.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                              |
| HasPerspective (string Perspective)                                   | String                                                                            | <b>Deprecated</b> - no longer in use.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                              |
| ImportPackageBuildScripts (string PackageGuid, string BuildScriptXML) |                                                                                   | Imports build scripts into a package in Enterprise Architect.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGuid: String - the GUID of the package into which to import the build scripts.</li> <li>BuildScriptXML: String - the build script XML data, which you can export from within Enterprise Architect.</li> </ul>                                                                                                                                                                                                                                           |                                                                                                                              |
| ImportTechnology (string Technology)                                  | Boolean                                                                           | Installs a given MDG Technology resource into the repository.<br>Returns <b>True</b> , if the technology is successfully loaded into the model. Otherwise returns <b>False</b> .<br>This applies to technologies imported into pre-7.0 versions of Enterprise Architect (imported technologies), not to technologies referenced in version 7.0 and later (referenced technologies).<br><b>Parameters:</b> <ul style="list-style-type: none"> <li>Technology: String - the contents of the technology resource file.</li> </ul>                                                     | <a href="#">Deploy an MDG Technology</a>  |
| IsTabOpen (string TabName)                                            | String                                                                            | Checks whether a named Enterprise Architect tabbed view is open and active. This includes open diagram windows or custom controls added using <b>Repository.AddTab()</b> .<br>Returns: <ul style="list-style-type: none"> <li><b>2</b> to indicate that a tab is open and active (top-most)</li> <li><b>1</b> to indicate that it is open but not top-most, or</li> <li><b>0</b> to indicate that it is not visible at all.</li> </ul> Parameters: <ul style="list-style-type: none"> <li>TabName: String - the name of the tab to check for; TabName is case sensitive</li> </ul> | <a href="#">Repository.AddTab()</a>     |
| IsTechnologyEnabled (string ID)                                       | Boolean                                                                           | Checks whether a specified technology is enabled in Enterprise Architect.<br>Returns <b>True</b> if the MDG Technology resource is enabled. Otherwise returns <b>False</b> .<br>Parameters: <ul style="list-style-type: none"> <li>ID: String - the technology ID to check for.</li> </ul>                                                                                                                                                                                                                                                                                         |                                                                                                                              |
| IsTechnologyLoaded (string ID)                                        | Boolean                                                                           | Checks whether a specified technology is loaded into the repository.<br>Returns <b>True</b> if the MDG Technology resource is loaded into the repository. Otherwise returns <b>False</b> .                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                              |

| Method                                                               | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See Also                                                         |
|----------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
|                                                                      |                | Parameters: <ul style="list-style-type: none"> <li>ID: String - the technology ID to check for.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                  |
| <b>OpenDiagram (long DiagramID)</b>                                  |                | Provides a method for an automation client or Add-In to open a diagram. The diagram is added to the tabbed list of open diagrams in the main Enterprise Architect view.<br><br>Parameters: <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to open.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                  |
| <b>OpenFile (string Filename)</b>                                    | <i>Boolean</i> | This is the main point for opening an Enterprise Architect project file from an automation client, and working with the contained objects.<br><br>If the required project is a DBMS repository, and you have created a shortcut .EAP file containing the database connection string, you can call this shortcut file to access the DBMS repository.<br><br>You can also connect to a SQL database by passing in the connection string itself instead of a filename. A valid connection string can be obtained from the <b>Open Project</b> dialog by selecting a recently opened SQL repository.<br><br>Parameters: <ul style="list-style-type: none"> <li>Filename: String - the filename of the Enterprise Architect project to open.</li> </ul> | <a href="#">Open a Project</a> <sup>[138]</sup>                  |
| <b>OpenFile2 (string FilePath, string Username, string Password)</b> | <i>Boolean</i> | As for <i>OpenFile()</i> except this enables the specification of a password.<br><br>Parameters: <ul style="list-style-type: none"> <li>Filepath: String - the file path of the Enterprise Architect project to open.</li> <li>Username: String - the user login ID</li> <li>Password: String - the user password.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                  |
| <b>RefreshModelView (long PackageID)</b>                             |                | Reloads a package or the entire model, updating the user interface.<br><br>Parameters: <ul style="list-style-type: none"> <li>PackageID: Long - the ID of the package to reload: if <b>0</b>, the entire model is reloaded; if a valid package ID, only that package is reloaded.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                  |
| <b>RefreshOpenDiagrams (boolean FullReload)</b>                      |                | Refreshes the diagram contents for all diagrams open in Enterprise Architect.<br><br>Parameters: <ul style="list-style-type: none"> <li>FullReload: Boolean - if <b>false</b> the displayed contents of elements and connectors are refreshed in each diagram; if <b>true</b> each of the diagrams is completely reloaded from the repository.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                          |                                                                  |
| <b>ReloadDiagram (long DiagramID)</b>                                |                | Reloads a specified diagram. This would commonly be used to refresh a visible diagram after code                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <a href="#">EA OnNotifyContentItemModified</a> <sup>[1992]</sup> |

| Method                                                                                                   | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See Also                                     |
|----------------------------------------------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
|                                                                                                          |                | <p>import/export or other batch process where the diagram requires complete refreshing.</p> <ul style="list-style-type: none"> <li>Calling this method within a call to <b>EA_OnNotifyContextItemModified</b> is not supported</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to be reloaded.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                         |                                              |
| <b>RemoveOutputTab (string Name)</b>                                                                     |                | <p>Removes a specified tab from the <b>Output</b> window.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Name: String - the name of the tab to be removed</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                              |
| <b>RunModelSearch (string sQueryName, string sSearchTerm, string sSearchOptions, string sSearchData)</b> |                | <p>Runs a search, displaying the results in Enterprise Architect's Model Search window</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>sQueryName: String - the name of the search to run, for example <i>Simple</i></li> <li>sSearchTerm: String - the term to search for</li> <li>sSearchOptions: String - currently not being used</li> <li>sSearchData: String - enables you to supply a list of results in the form of XML, which is appended onto the result list in Enterprise Architect - see <b>XML Format</b>; this parameter is not mandatory so pass in an empty string to run the search as per normal</li> </ul>                                                                                                   | <a href="#">XML Format</a> <sup>[1975]</sup> |
| <b>SaveAllDiagrams ()</b>                                                                                |                | Saves all open diagrams                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                              |
| <b>SaveAuditLogs (string FilePath, object StartDateTime, object EndDateTime)</b>                         | <i>Boolean</i> | <p>Saves the Audit Logs contained within a model to a specified file</p> <p>If <i>StartDateTime</i> and <i>EndDateTime</i> are not null then only log items that fall into this period are saved</p> <p>Returns <b>true</b> for success, <b>false</b> for failure</p> <ul style="list-style-type: none"> <li>This might fail if the user logged into the model does not have the correct access permission</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>FilePath: String - the file to save the Audit Logs to</li> <li>StartDateTime: Variant ( DateTime ) - the earliest date and time of log entries to save</li> <li>EndDateTime; Variant ( DateTime ) - the latest date and time of log entries to save</li> </ul> |                                              |
| <b>SaveDiagram (long DiagramID)</b>                                                                      |                | <p>Saves an open diagram; assumes the diagram is open in the main user interface Tab list</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to save</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                              |

| Method                                 | Type | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also |
|----------------------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>ScanXMIAndReconcile()</b>           |      | <p>Scans the package XML files associated with each of the project's controlled packages and restores any diagram objects or cross-references that are detected as missing from the project</p> <p>This function is useful in team environments where each user maintains their own private copy of the model database (i.e. multiple private EAP files) and model updates are propagated through the use of controlled packages; it provides no benefit when the model is hosted in a single shared database that is accessed by all team members</p> <p>Each controlled package is compared with its associated XML file and, if the cross-reference information in the model does not match the XML, Enterprise Architect updates the model with the information from the XML and records the update in the Output window</p> <p>You can roll back such updates by right-clicking on the entry in the Output window and selecting the context menu option <b>Rollback Update</b> (or <b>Rollback Selected Updates</b> if multiple entries are selected)</p> <p>Closing the model clears the entries in the Output window; an entry in the Output window is also cleared as and when you roll-back the update for it</p> <p>This functionality is invoked automatically as part of the <b>Get All Latest</b> operation</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke this function from time to time, following the re-importation of controlled packages - for example, after using <b>Get Latest</b> to update a number of packages, or after performing a number of package check-outs</p> <ul style="list-style-type: none"> <li>• As a general rule, avoid running this function while you have uncommitted changes in your model</li> <li>• Generally, you should: <ul style="list-style-type: none"> <li>• Check-out a number of packages</li> <li>• Invoke <b>ScanXMIAndReconcile</b></li> <li>• Make your modifications</li> <li>• Commit any outstanding changes before you check-out more packages and run <b>ScanXMIAndReconcile</b> again</li> </ul> </li> </ul> |          |
| <b>ShowDynamicHelp (string Topic)</b>  |      | <p>Shows a help topic as a view</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Topic: String - specifies the help topic</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |          |
| <b>ShowInProjectView (object Item)</b> |      | <p>Selects a specified object in the Project Browser</p> <p>Accepted object types are <i>Package</i>, <i>Element</i>, <i>Diagram</i>, <i>Attribute</i>, and <i>Method</i>; an exception is</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |          |

| Method                                                                      | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See Also                                                                                      |
|-----------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
|                                                                             |                | thrown if the object is of an invalid type<br>Parameters:<br><ul style="list-style-type: none"> <li>Item: Object - the object to highlight</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                               |
| <b>ShowProfileToolbox (string Technology, string Profile, boolean Show)</b> |                | Shows/hides the contents of a specified technology or profile in the Toolbox<br><br>To show/hide a profile in the Toolbox, specify the profile's ID value in the <i>Profile</i> parameter and set the <i>Technology</i> parameter to a null string<br><br>To show/hide a technology in the Toolbox, specify the technology's ID in the <i>Technology</i> parameter and set the <i>Profile</i> parameter to a null string<br>Parameters:<br><ul style="list-style-type: none"> <li>Technology: String - the ID of the technology</li> <li>Profile: String - the ID of the profile</li> <li>Show: Boolean - if <b>true</b>, show the technology or profile; if <b>false</b>, hide the technology or profile</li> </ul> |                                                                                               |
| <b>ShowWindow (long Show)</b>                                               |                | Shows or hides Enterprise Architect<br>Parameters:<br><ul style="list-style-type: none"> <li>Show: Long</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                               |
| <b>SQLQuery (string SQL)</b>                                                | <i>String</i>  | Enables execution of a SQL <i>select</i> statement against the current repository<br><br>Returns an XML formatted string value of the resulting recordset<br>Parameters:<br><ul style="list-style-type: none"> <li>SQL: String - contains the SQL Select statement</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                               |
| <b>SynchProfile (string Profile, string Stereotype)</b>                     | <i>Boolean</i> | Synchronizes Tagged Values and constraints of a UML Profile item using the Synch Profile Elements dialog<br>Parameters:<br><ul style="list-style-type: none"> <li>Profile: String - the name of the profile that contains the stereotype</li> <li>Stereotype: String - the name of the profile stereotype for which the default tags and constraints are to be synchronized</li> </ul>                                                                                                                                                                                                                                                                                                                               | <a href="#">Synchronize Tagged Values and Constraints</a><br><small>1029</small>              |
| <b>VersionControlResynchPkgStatuses (boolean ClearSettings)</b>             |                | <b>Synchronizes the version control status</b> of each version controlled package within the current model with the status reported by your version control provider<br>Parameters:<br><ul style="list-style-type: none"> <li>ClearSettings: Boolean - if <b>true</b>, clear the version control settings from packages that are reported by the version control provider as <i>uncontrolled</i>; if <b>false</b>, leave the version control settings unchanged for packages</li> </ul>                                                                                                                                                                                                                              | <a href="#">Resynchronize the Status of Version Controlled Packages</a><br><small>297</small> |

| Method                                                   | Type | Notes                                                                                                                                                                                                                                                                                                                                                          | See Also                                                                                                                                                    |
|----------------------------------------------------------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                          |      | reported as <i>uncontrolled</i>                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                             |
| <b>WriteOutput (string Name, string String, long ID)</b> |      | Writes text to a specified tab in the <b>Output</b> window, and associates the text with an ID.<br><br>Parameters: <ul style="list-style-type: none"> <li>Name: String - specifies the tab on which to display the text.</li> <li>String: String - specifies the text to display.</li> <li>ID: Long - specifies the ID the text is associated with.</li> </ul> | <a href="#">ClearOutput</a> <sup>[1859]</sup><br><a href="#">CreateOutputTab</a> <sup>[1860]</sup><br><a href="#">EnsureOutputVisible</a> <sup>[1860]</sup> |

#### 18.2.2.4.15 Stereotype Class

The *Stereotype* element corresponds to a UML stereotype, which is an extension mechanism for varying the behavior and type of a model element. Use the Repository *Stereotypes* collection to add new elements and delete existing ones.

Associated table in .EAP file: *t\_stereotypes*

#### Stereotype Attributes

| Attribute               | Type                                         | Notes                                                                                                                                                                                                                  |
|-------------------------|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AppliesTo</b>        | <i>String</i>                                | Read/Write. A reference to the stereotype <i>Base Class</i> , that is, which element it applies to.                                                                                                                    |
| <b>MetafileLoadPath</b> | <i>String</i>                                | Read/Write. Path to an associated metafile. The automation interface does not yet support loading metafiles. To do this you must use the <b>Stereotype</b> tab of the <b>UML Types</b> dialog in Enterprise Architect. |
| <b>Notes</b>            | <i>String</i>                                | Read/Write. Notes about the stereotype.                                                                                                                                                                                |
| <b>Name</b>             | <i>String</i>                                | Read/Write. The stereotype name. Appears in the <b>Stereotype</b> drop list for elements that match the <i>AppliesTo</i> attribute.                                                                                    |
| <b>ObjectType</b>       | <a href="#">ObjectType</a> <sup>[1835]</sup> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                              |
| <b>StereotypeGUID</b>   | <i>String</i>                                | Read/Write. Unique identifier for stereotype, generally set and maintained by Enterprise Architect.                                                                                                                    |
| <b>Style</b>            | <i>String</i>                                | Read/Write. Additional style specifier for stereotype.                                                                                                                                                                 |
| <b>VisualType</b>       | <i>String</i>                                | Read/Write. Indicates an inbuilt visual style associated with a stereotype. Not currently implemented.                                                                                                                 |

#### Stereotype Methods

| Method                 | Type          | Notes                                                                                                                                                                         |
|------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error |

| Method           | Type           | Notes                                                                                                                                                                      |
|------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  |                | occurs.                                                                                                                                                                    |
| <b>Update ()</b> | <i>Boolean</i> | Update the current stereotype object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 18.2.2.4.16 Task Class

A Task is an entry in the System ToDo list. Accessed using the Repository *Tasks* collection.

Associated table in .EAP file: *t\_tasks*

#### Task Attributes

| Attribute         | Type                                 | Notes                                                                                       |
|-------------------|--------------------------------------|---------------------------------------------------------------------------------------------|
| <b>ActualTime</b> | <i>Long</i>                          | Read/Write. Time already expended on task, in hours, days or other units.                   |
| <b>AssignedTo</b> | <i>String</i>                        | Read/Write. Person this task is assigned to; that is, the responsible resource.             |
| <b>EndDate</b>    | <i>Date</i>                          | Read/Write. Date task scheduled to finish.                                                  |
| <b>History</b>    | <i>String</i>                        | Read/Write. Memo field to hold, for example, task history or notes.                         |
| <b>Name</b>       | <i>Variant</i>                       | Read/Write. Task name.                                                                      |
| <b>Notes</b>      | <i>Variant</i>                       | Read/Write. Description of the task.                                                        |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.                   |
| <b>Owner</b>      | <i>String</i>                        | Read/Write. The task owner.                                                                 |
| <b>Percent</b>    | <i>Long</i>                          | Read/Write. Percent the task is complete.                                                   |
| <b>Phase</b>      | <i>String</i>                        | Read/Write. The phase of the project the task relates to.                                   |
| <b>Priority</b>   | <i>String</i>                        | Read/Write. Priority associated with this task.                                             |
| <b>StartDate</b>  | <i>Date</i>                          | Read/Write. Date task is to start.                                                          |
| <b>Status</b>     | <i>Variant</i>                       | Read/Write. Current task status.                                                            |
| <b>TaskID</b>     | <i>Long</i>                          | Read only. Local ID of task.                                                                |
| <b>TotalTime</b>  | <i>Long</i>                          | Read/Write. The total expected time the task might run - in hours, days or some other unit. |
| <b>Type</b>       | <i>String</i>                        | Read/Write. Sets or returns string representing the type.                                   |

#### Task Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Task object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                  |

#### 18.2.2.4.17 Term Class

A *Term* object represents one entry in the system glossary. Accessed using the Repository *Terms* collection.

Associated table in .EAP file: *t\_glossary*

#### Term Attributes

| Attribute         | Type                                 | Notes                                                                           |
|-------------------|--------------------------------------|---------------------------------------------------------------------------------|
| <b>Meaning</b>    | <i>String</i>                        | Read/Write. The description of the term; its meaning.                           |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.       |
| <b>Term</b>       | <i>String</i>                        | Read/Write. The glossary item name.                                             |
| <b>TermID</b>     | <i>Long</i>                          | Read only. A local ID number to identify the term in the model.                 |
| <b>Type</b>       | <i>String</i>                        | Read/Write. The type this term applies to (for example, business or technical). |

#### Term Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Term object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                  |

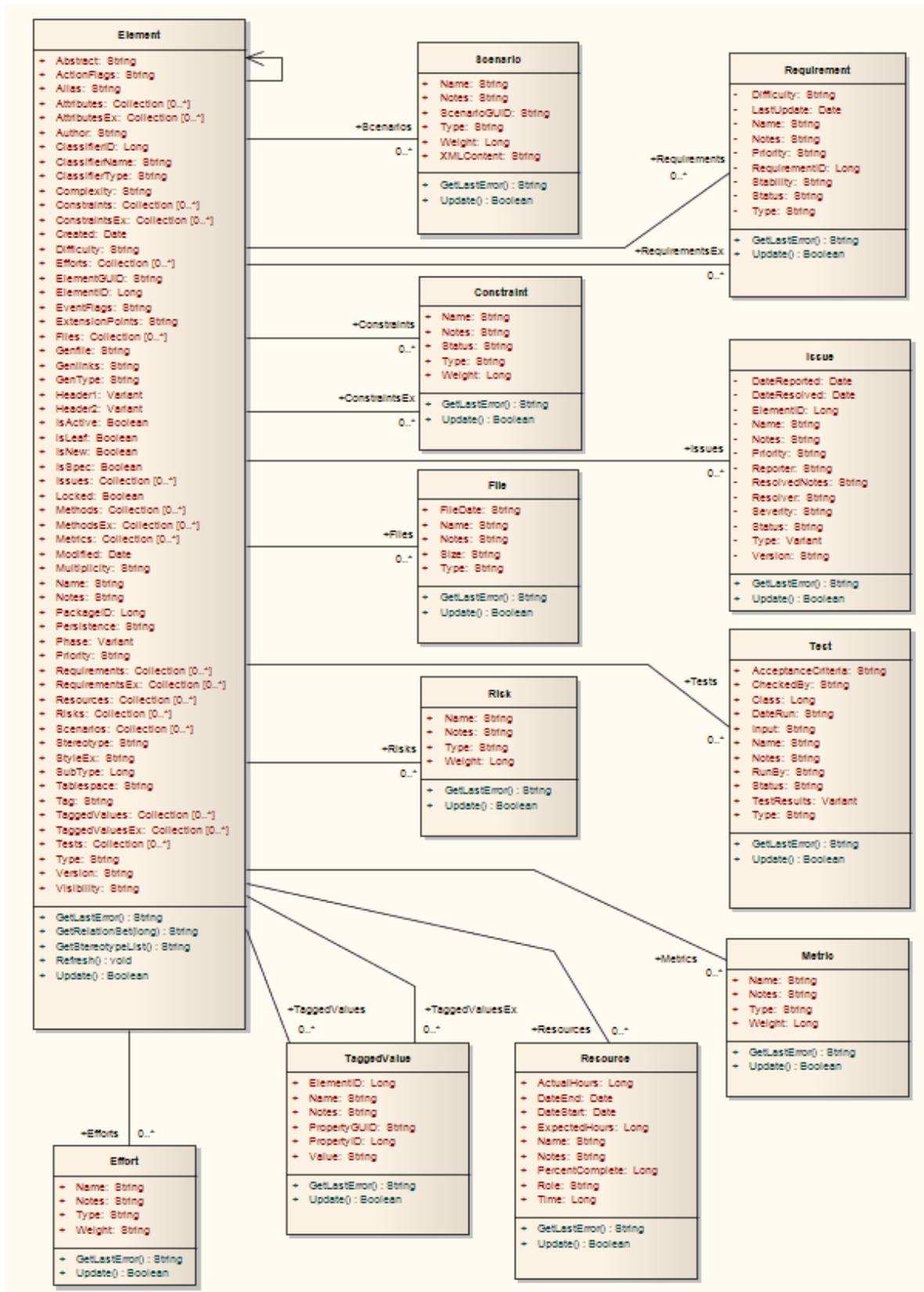
#### 18.2.2.5 Element Package

The *Element* package contains information about an element and its associated extended properties such as testing and project management information. An element is the basic item in an Enterprise Architect model. Classes, Use Cases and Components are all different types of UML element.

The diagram below illustrates the relationships between an *element* and its associated extended information. The related information is accessed through the collections owned by the element (for example, Scenarios and Tests). It also includes a full description of the element object (the basic model structural

unit).

**Example:**



### 18.2.2.5.1 Constraint Class

A *Constraint* is a condition imposed on an element. Constraints are accessed through the Element *Constraints* collection.

Associated table in .EAP file: *t\_objectconstraints*

#### Constraint Attributes

| Attribute         | Type                                 | Notes                                                                            |
|-------------------|--------------------------------------|----------------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                        | Read/Write. The name of the constraint (that is, the constraint).                |
| <b>Notes</b>      | <i>String</i>                        | Read/Write. Notes about the constraint.                                          |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.        |
| <b>ParentID</b>   | <i>Long</i>                          | Read only. The <i>ElementID</i> of the element to which this constraint applies. |
| <b>Status</b>     | <i>String</i>                        | Read/Write. Current status.                                                      |
| <b>Type</b>       | <i>String</i>                        | Read/Write. Constraint type.                                                     |
| <b>Weight</b>     | <i>Long</i>                          | Read/Write. A weighting factor.                                                  |

#### Constraint Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>Constraint</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.     |

### 18.2.2.5.2 Effort Class

An *Effort* is a named item with a weighting that can be associated with an element for purposes of building metrics about the model. Accessed through the Element *Efforts* collection.

Associated table in .EAP file: *t\_objecteffort*

#### Effort Attributes

| Attribute    | Type          | Notes                               |
|--------------|---------------|-------------------------------------|
| <b>Name</b>  | <i>String</i> | Read/Write. The name of the effort. |
| <b>Notes</b> | <i>String</i> | Read/Write. Notes about the effort. |

| Attribute         | Type                                              | Notes                                                                     |
|-------------------|---------------------------------------------------|---------------------------------------------------------------------------|
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>1835</small> | Read only. Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                                     | Read/Write. The effort type.                                              |
| <b>Weight</b>     | <i>Long</i>                                       | Read/Write. A weighting factor.                                           |
| <b>Weight2</b>    | <i>Float</i>                                      | Read/Write. A weighting factor.                                           |

#### Effort Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Saves the effort to the model.                                                                                                                                                        |

#### 18.2.2.5.3 Element Class

An *Element* is the main modeling unit. It corresponds to (for example) Class, Use Case, Node or Component. You create new elements by adding to the Package *Elements* collection. Once you have created an element, you can add it to the *DiagramObject* Class of a diagram to include it in the diagram.

Elements also have a collection of connectors. Each entry in this collection indicates a relationship to another element.

There are also some extended collections for managing additional information about the element, including things such as Tagged Values, Issues, Constraints and Requirements.

Associated table in .EAP file: *t\_object*

#### Element Attributes

| Attribute                          | Type          | Notes                                                                                               | See Also                                                                           |
|------------------------------------|---------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <b>Abstract</b>                    | <i>String</i> | Read/Write<br>Indicates if the element is Abstract (1) or Concrete (0)                              |                                                                                    |
| <b>ActionFlags</b>                 | <i>String</i> | Read/Write<br>A structure to hold flags concerned with Action semantics                             |                                                                                    |
| <b>Alias</b>                       | <i>String</i> | Read/Write<br>An optional alias for this element                                                    |                                                                                    |
| <b>AssociationClassConnectorID</b> | <i>Long</i>   | Read only<br>If the element is an AssociationClass, <i>AssociationClassConnectorID</i> contains the | <a href="#">CreateAssociationClass()</a><br><small>1885</small><br>(Information on |

| Attribute               | Type                                                | Notes                                                                                                                                                                                                        | See Also                                                     |
|-------------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
|                         |                                                     | Connector ID of the respective Association connector                                                                                                                                                         | creating AssociationClasses from elements)                   |
| <b>Attributes</b>       | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of Attribute objects for the current element; use the <b>AddNew</b> and <b>Delete</b> functions to manage attributes                                                                 |                                                              |
| <b>AttributesEx</b>     | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of Attribute objects belonging to the current element and its parent elements                                                                                                        |                                                              |
| <b>Author</b>           | String                                              | Read/Write<br>The element author                                                                                                                                                                             | <a href="#">Repository: Authors</a><br><small>[1855]</small> |
| <b>BaseClasses</b>      | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>List of Base Classes for this element, presented as a collection for convenience                                                                                                                |                                                              |
| <b>ClassifierID</b>     | Long                                                | <b>Deprecated</b><br>See <i>ClassifierID</i>                                                                                                                                                                 |                                                              |
| <b>ClassifierID</b>     | Long                                                | Read/Write<br>The ElementID of a Classifier associated with this element; that is, the base type<br><br>Only valid for instance type elements (such as Object or Sequence)                                   |                                                              |
| <b>ClassifierName</b>   | String                                              | Read/Write<br>Name of associated Classifier (if any)                                                                                                                                                         |                                                              |
| <b>ClassifierType</b>   | String                                              | Read only<br>Type of associated Classifier                                                                                                                                                                   |                                                              |
| <b>Complexity</b>       | String                                              | Read/Write<br><br>A complexity value indicating how complex the element is; used for metric reporting and estimation<br><br>Valid values are: <b>1</b> for Easy, <b>2</b> for Medium, <b>3</b> for Difficult |                                                              |
| <b>CompositeDiagram</b> | <a href="#">Diagram</a><br><small>[1927]</small>    | Read only<br>If the element is Composite, returns its associated diagram; otherwise returns null                                                                                                             |                                                              |
| <b>Connectors</b>       | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Returns a collection containing the connectors to other elements                                                                                                                                |                                                              |
| <b>Constraints</b>      | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of <b>Constraint</b> objects                                                                                                                                                         | <a href="#">Constraint Class</a> <small>[1876]</small>       |

| Attribute        | Type                                                | Notes                                                                                                                                                                                                                                                            | See Also                                              |
|------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| ConstraintsEx    | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of Constraint objects belonging to the current element and its parent elements                                                                                                                                                           |                                                       |
| Created          | Date                                                | Read/Write<br>The date the element was created                                                                                                                                                                                                                   |                                                       |
| CustomProperties | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>List of advanced properties for an element<br>The collection of advanced properties differs depending on element type; for example, an Action and an Activity have different advanced properties<br>Currently only editable from the user interface |                                                       |
| Diagrams         | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Returns a collection of sub-diagrams (child diagrams) attached to this element as seen in the tree view                                                                                                                                             |                                                       |
| Difficulty       | String                                              | Read/Write<br>A difficulty level associated with this element for estimation/metrics; only useable for Requirement, Change and Issue element types, otherwise ignored<br>Valid values are: <b>Low</b> , <b>Medium</b> , <b>High</b>                              |                                                       |
| Efforts          | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of <b>Effort</b> objects                                                                                                                                                                                                                 | <a href="#">Effort Class</a><br><small>[1876]</small> |
| ElementGUID      | String                                              | Read only<br>A globally unique ID for this element; that is, unique across all model files                                                                                                                                                                       |                                                       |
| ElementID        | Long                                                | Read only<br>The local ID of the Element; valid for this file only                                                                                                                                                                                               |                                                       |
| Elements         | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Returns a collection of child elements (sub-elements) attached to this element as seen in the tree view                                                                                                                                             |                                                       |
| EmbeddedElements | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>List of elements that are embedded into this element, such as Ports, Parts, Pins and Parameter Sets                                                                                                                                                 |                                                       |
| EventFlags       | String                                              | Read/Write<br>A structure to hold a variety of flags to do with signals or events                                                                                                                                                                                |                                                       |
| ExtensionPoints  | String                                              | Read/Write                                                                                                                                                                                                                                                       |                                                       |

| Attribute          | Type                                                | Notes                                                                                                                                                                    | See Also                                            |
|--------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
|                    |                                                     | Optional extension points for a Use Case as a comma-separated list                                                                                                       |                                                     |
| <b>Files</b>       | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of <b>File</b> objects                                                                                                                           | <a href="#">File Class</a><br><small>[1887]</small> |
| <b>GenFile</b>     | <i>String</i>                                       | Read/Write<br>The file associated with this element for code generation and synchronization purposes; can include macro expansion tags for local conversion to full path |                                                     |
| <b>Genlinks</b>    | <i>String</i>                                       | Read/Write<br>Links to other Classes discovered at code reversing time; Parents and Implements connectors only                                                           |                                                     |
| <b>GenType</b>     | <i>String</i>                                       | Read/Write<br>The code generation type; for example, Java, C++, C#, VBNet, Visual Basic, Delphi                                                                          |                                                     |
| <b>Header1</b>     | <i>Variant</i>                                      | Read/Write<br>A user defined string for inclusion as header in the source files generated                                                                                |                                                     |
| <b>Header2</b>     | <i>Variant</i>                                      | Read/Write<br>Same as for <b>Header1</b> , but used in the CPP source file                                                                                               |                                                     |
| <b>IsActive</b>    | <i>Boolean</i>                                      | Read/Write<br>Boolean value indicating whether the element is active or not<br><b>1</b> = True, <b>0</b> = False                                                         |                                                     |
| <b>IsComposite</b> | <i>Boolean</i>                                      | Read/Write<br>Indicates whether the element is composite or not<br><b>1</b> = True, <b>0</b> = False                                                                     |                                                     |
| <b>IsLeaf</b>      | <i>Boolean</i>                                      | Read/Write<br>Boolean value indicating whether the element is in leaf node or not<br><b>1</b> = True, <b>0</b> = False                                                   |                                                     |
| <b>IsNew</b>       | <i>Boolean</i>                                      | Read/Write<br>Boolean value indicating whether the element is new or not<br><b>1</b> = True, <b>0</b> = False                                                            |                                                     |
| <b>IsSpec</b>      | <i>Boolean</i>                                      | Read/Write<br>Boolean value indicating whether the element is                                                                                                            |                                                     |

| Attribute    | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also |
|--------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|              |                                                     | a specification or not<br>1 = True, 0 = False                                                                                                                                                                                                                                                                                                                                                                                                                            |          |
| Issues       | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of Issue objects                                                                                                                                                                                                                                                                                                                                                                                                                                 |          |
| Locked       | Boolean                                             | Read/Write<br>Indicates if the element has been locked against further change                                                                                                                                                                                                                                                                                                                                                                                            |          |
| MetaType     | String                                              | Read only<br>The element's domain-specific meta type, as defined by an applied stereotype from an MDG Technology                                                                                                                                                                                                                                                                                                                                                         |          |
| Methods      | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of Method objects for current element                                                                                                                                                                                                                                                                                                                                                                                                            |          |
| MethodsEx    | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of Method objects belonging to the current element and its parent elements                                                                                                                                                                                                                                                                                                                                                                       |          |
| Metrics      | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of Metric elements for current element                                                                                                                                                                                                                                                                                                                                                                                                           |          |
| MiscData     | String                                              | Read only<br>This low-level property provides information about the contents of the <b>PData</b> x fields<br>These database fields are not documented, and developers must gain understanding of these fields through their own endeavors to use this property<br><b>MiscData</b> is zero based, therefore: <ul style="list-style-type: none"> <li>• <b>MiscData(0)</b> corresponds to <b>PData1</b></li> <li>• <b>MiscData(1)</b> to <b>PData2</b></li> </ul> and so on |          |
| Modified     | Date                                                | Read/Write<br>The date the element was last modified                                                                                                                                                                                                                                                                                                                                                                                                                     |          |
| Multiplicity | String                                              | Read/Write<br>Multiplicity value for this element                                                                                                                                                                                                                                                                                                                                                                                                                        |          |
| Name         | String                                              | Read/Write<br>The element name; should be unique within the current package                                                                                                                                                                                                                                                                                                                                                                                              |          |
| Notes        | String                                              | Read/Write<br>Further descriptive text about the element                                                                                                                                                                                                                                                                                                                                                                                                                 |          |
| ObjectType   | <a href="#">ObjectTyp</a>                           | Read only                                                                                                                                                                                                                                                                                                                                                                                                                                                                |          |

| Attribute      | Type                                       | Notes                                                                                                                                                                                                                                                                                          | See Also                                          |
|----------------|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
|                | <a href="#">e</a> <sup>1835</sup>          | Distinguishes objects referenced through a Dispatch interface                                                                                                                                                                                                                                  |                                                   |
| PackageID      | Long                                       | Read/Write<br>A local ID for the package containing this element                                                                                                                                                                                                                               |                                                   |
| ParentID       | Long                                       | Read/Write<br>If this element is a child of another, used to set or retrieve the <i>ElementID</i> of the other element; if not, returns <b>0</b>                                                                                                                                               |                                                   |
| Partitions     | <a href="#">Collection</a> <sup>1847</sup> | Read only<br>List of logical partitions into which an element can be divided<br><br>Only valid for elements that support partitions, such as Activities and States                                                                                                                             |                                                   |
| Persistence    | String                                     | Read/Write<br>The persistence associated with this element; can be <b>Persistent</b> or <b>Transient</b>                                                                                                                                                                                       |                                                   |
| Phase          | String                                     | Read/Write<br>The phase this element is scheduled to be constructed in; any string value                                                                                                                                                                                                       |                                                   |
| Priority       | String                                     | Read/Write<br>The priority of this element as compared to other project elements; only applies to Requirement, Change and Issue types, otherwise ignored<br><br>Valid values are: <b>Low</b> , <b>Medium</b> and <b>High</b>                                                                   |                                                   |
| Properties     | <a href="#">Properties</a> <sup>1909</sup> | Returns a list of specialized properties that apply to the element that might not be available using the automation model<br><br>The properties are purposely undocumented because of their obscure nature and because they are subject to change as progressive enhancements are made to them |                                                   |
| PropertyType   | Long                                       | Read/Write<br>The ElementID of a Type associated with this element; only valid for Port and Part elements                                                                                                                                                                                      |                                                   |
| Realizes       | <a href="#">Collection</a> <sup>1847</sup> | Read only<br>List of Interfaces realized by this element for convenience                                                                                                                                                                                                                       |                                                   |
| Requirements   | <a href="#">Collection</a> <sup>1847</sup> | Read only<br>Collection of <b>Requirement</b> objects                                                                                                                                                                                                                                          | <a href="#">Requirement Class</a> <sup>1890</sup> |
| RequirementsEx | <a href="#">Collection</a> <sup>1847</sup> | Read only<br>Collection of <b>Requirement</b> objects belonging to the current element and its parent elements                                                                                                                                                                                 | <a href="#">Requirement Class</a> <sup>1890</sup> |

| Attribute        | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See Also                                                |
|------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| Resources        | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of <b>Resource</b> objects for current element                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <a href="#">Resource Class</a><br><small>[1891]</small> |
| Risks            | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of <b>Risk</b> objects                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <a href="#">Risk Class</a><br><small>[1891]</small>     |
| RunState         | String                                              | Read/Write<br>The object's runstate list as a string                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                         |
| Scenarios        | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>Collection of <b>Scenario</b> objects for current element                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <a href="#">Scenario Class</a><br><small>[1892]</small> |
| StateTransitions | <a href="#">Collection</a><br><small>[1847]</small> | Read only<br>List of State Transitions that an element can support; applies in particular to Timing elements                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                         |
| Status           | String                                              | Read/Write<br>Sets or gets the status, such as <b>Proposed</b> or <b>Approved</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                         |
| Stereotype       | String                                              | Read/Write<br>The primary element stereotype; the first of the list of stereotypes you can access using the <i>StereotypeEx</i> attribute                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                         |
| StereotypeEx     | String                                              | Read/Write<br>All the applied stereotypes of the element in a comma-separated list                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                         |
| StyleEx          | String                                              | Read/Write<br>Advanced style settings; reserved for the use of Sparx Systems                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                         |
| Subtype          | Long                                                | Read/Write<br>A numeric subtype that qualifies the <b>Type</b> of the main element; for example: <ul style="list-style-type: none"> <li>• For Event: <b>0</b> = Receiver, <b>1</b> = Sender</li> <li>• For Class: <b>1</b> = Parameterised, <b>2</b> = Instantiated, <b>3</b> = Both, <b>0</b> = Neither, <b>17</b> = Association Class</li> </ul> If <b>17</b> , because an Association Class has been created through the user interface, <b>MiscData(3)</b> contains the ID of the related Association; as MiscData is read-only, you cannot create an Association Class through the Automation Interface <ul style="list-style-type: none"> <li>• For Note: <b>1</b> = Note linked to connector, <b>2</b> = Constraint linked to connector</li> <li>• For StateNode: <b>100</b> = ActivityInitial, <b>101</b> = ActivityFinal</li> <li>• For Activity: <b>0</b> = Activity, <b>8</b> = composite</li> </ul> | <a href="#">Type</a><br><small>[1884]</small>           |

| Attribute             | Type                                                                                               | Notes                                                                                                                                                                                                                                                                                                                                                                                                              | See Also                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------|----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       |                                                                                                    | <p>Activity (also set to <b>8</b> for other composite elements such as Use Cases)</p> <ul style="list-style-type: none"> <li>For Synchronization: <b>0</b> = Horizontal, <b>1</b> = Vertical</li> </ul> <p>Note that there are many more Types than indicated in the above examples</p>                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Tablespace</b>     | <i>String</i>                                                                                      | <p>Read/Write</p> <p>Associated tablespace for a Table element</p>                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Tag</b>            | <i>String</i>                                                                                      | <p>Read/Write</p> <p>Corresponds to the <b>Keywords</b> field in the Enterprise Architect user interface</p>                                                                                                                                                                                                                                                                                                       | <a href="#">General Settings</a> <sup>[662]</sup>                                                                                                                                                                                                                                                                                                                                                                      |
| <b>TaggedValues</b>   | <a href="#">Collection</a> <sup>[1847]</sup> of type <a href="#">TaggedValue</a> <sup>[1895]</sup> | <p>Read only</p> <p>Returns a collection of <b>TaggedValue</b> objects</p>                                                                                                                                                                                                                                                                                                                                         | <a href="#">TaggedValue Class</a> <sup>[1895]</sup>                                                                                                                                                                                                                                                                                                                                                                    |
| <b>TaggedValuesEx</b> | <a href="#">Collection</a> <sup>[1847]</sup> of type <a href="#">TaggedValue</a> <sup>[1895]</sup> | <p>Read only</p> <p>Returns a collection of <b>TaggedValue</b> objects belonging to the current element and the elements specialized or realized by the current element</p>                                                                                                                                                                                                                                        | <a href="#">TaggedValue Class</a> <sup>[1895]</sup>                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Tests</b>          | <a href="#">Collection</a> <sup>[1847]</sup>                                                       | <p>Read only</p> <p>Collection of <b>Test</b> objects for current element</p>                                                                                                                                                                                                                                                                                                                                      | <a href="#">Test Class</a> <sup>[1896]</sup>                                                                                                                                                                                                                                                                                                                                                                           |
| <b>TreePos</b>        | <i>Long</i>                                                                                        | <p>Read/Write</p> <p>Sets or gets the tree position</p>                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Type</b>           | <i>String</i>                                                                                      | <p>Read/Write</p> <p>The element type (such as Class, Component)</p> <p>Note that Type is case sensitive inside Enterprise Architect and should be provided with an initial capital (proper case); valid types are:</p>                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                       |                                                                                                    | <p><b>Action</b></p> <p><b>Activity</b></p> <p><b>ActivityPartition</b></p> <p><b>ActivityRegion</b></p> <p><b>Actor</b></p> <p><b>Artifact</b></p> <p><b>Association</b></p> <p><b>Boundary</b></p> <p><b>Change</b></p> <p><b>Class</b></p> <p><b>Collaboration</b></p> <p><b>Component</b></p> <p><b>Constraint</b></p> <p><b>Decision</b></p> <p><b>DeploymentSpecification</b></p> <p><b>DiagramFrame</b></p> | <p><b>InteractionOccurrence</b></p> <p><b>InteractionState</b></p> <p><b>Interface</b></p> <p><b>InterruptibleActivityRegion</b></p> <p><b>Issue</b></p> <p><b>Node</b></p> <p><b>Note</b></p> <p><b>Object</b></p> <p><b>Package</b></p> <p><b>Parameter</b></p> <p><b>Part</b></p> <p><b>Port</b></p> <p><b>ProvidedInterface</b></p> <p><b>Report</b></p> <p><b>RequiredInterface</b></p> <p><b>Requirement</b></p> |

| Attribute         | Type          | Notes                                                                                                                                                                                                                            | See Also                                                                                                                                                                |
|-------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   |               | <b>EmbeddedElement</b><br><b>Entity</b><br><b>EntryPoint</b><br><b>Event</b><br><b>ExceptionHandler</b><br><b>ExitPoint</b><br><b>ExpansionNode</b><br><b>ExpansionRegion</b><br><b>GUIElement</b><br><b>InteractionFragment</b> | <b>Screen</b><br><b>Sequence</b><br><b>State</b><br><b>StateNode</b><br><b>Synchronization</b><br><b>Text</b><br><b>TimeLine</b><br><b>UMLDiagram</b><br><b>UseCase</b> |
| <b>Version</b>    | <i>String</i> | Read/Write<br>The version of the element                                                                                                                                                                                         |                                                                                                                                                                         |
| <b>Visibility</b> | <i>String</i> | Read/Write<br>The Scope of this element within the current package<br>Valid values are: <b>Public</b> , <b>Private</b> , <b>Protected</b> or <b>Package</b>                                                                      |                                                                                                                                                                         |

### Element Methods

| Method                                           | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also |
|--------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>ApplyGroupLock (string aGroupName)</b>        | <i>Boolean</i> | Applies a group lock to the element object, for the specified group, on behalf of the current user<br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information<br>Parameter: <ul style="list-style-type: none"> <li>aGroupName: String - the name of the user group for which to set the group lock</li> </ul>                                                                                                                                                                                    |          |
| <b>ApplyUserLock ()</b>                          | <i>Boolean</i> | Applies a user lock to the element object for the current user<br>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information                                                                                                                                                                                                                                                                                                                                                                        |          |
| <b>CreateAssociationClass (long ConnectorID)</b> | <i>Boolean</i> | Makes this element an AssociationClass of the Association with the provided <b>Connector ID</b> ; the return value indicates whether the function succeeded in converting the element to an AssociationClass<br>AssociationClasses are created only where: <ol style="list-style-type: none"> <li>The current element is valid</li> <li>The current element is a Class</li> <li>The current element is not already an AssociationClass</li> <li>The specified connector exists</li> <li>The specified connector is an Association</li> </ol> |          |

| Method                                           | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See Also                                                                                             |
|--------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
|                                                  |                | <p>6. The specified connector is not already in an AssociationClass pair</p> <p>7. The current element is not at either end of the specified connector</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>ConnectorID: Long - the Connector ID of an Association connector</li> </ul>                                                                                                                                                                                                                                                                 |                                                                                                      |
| <b>GetLastError ()</b>                           | <i>String</i>  | <p>Returns a string value describing the most recent error that occurred in relation to this object</p> <p>This function is rarely used as an exception is thrown when an error occurs</p>                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                      |
| <b>GetLinkedDocument ()</b>                      | <i>String</i>  | <p>Returns a string value containing the element's linked document contents, in RTF format</p> <p>If the element contains no linked document, an empty string is returned</p>                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                      |
| <b>GetRelationSet (EnumRelationSetType Type)</b> | <i>String</i>  | <p>Returns a string containing a comma-separated list of ElementIDs of directly- and indirectly-related elements based on the given type</p> <p>Recurses using the same relation type on all elements it finds, retrieving all dependencies and sub-dependencies of the current element; for example, <i>Object1</i> depends on <i>Object2</i>, which depends on <i>Object3</i>, therefore this method returns <i>Object2</i> and <i>Object3</i></p> <p>To obtain only the direct relationships of the element, use the <i>Connector</i> collection instead</p> | <a href="#">EnumRelationSetType</a> <sup>[1834]</sup><br><a href="#">Connector</a> <sup>[1910]</sup> |
| <b>GetStereotypeList ()</b>                      | <i>String</i>  | Returns a comma-separated list of stereotypes allied to this element                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                      |
| <b>IsAssociationClass</b>                        | <i>Boolean</i> | Returns whether or not the current element is an AssociationClass                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                      |
| <b>LoadLinkedDocument (string Filename)</b>      | <i>Boolean</i> | <p>Loads the RTF document from the specified file into the element's linked document</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>FileName: String - the name of the file from which to load the RTF document</li> </ul>                                                                                                                                                                                                                                                                                                                        |                                                                                                      |
| <b>Refresh ()</b>                                | <i>Void</i>    | <p>Refreshes the element features in the Project Browser</p> <p>Usually called after adding or deleting attributes or methods, when the user interface is required to be updated as well</p>                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                      |
| <b>ReleaseUserLock ()</b>                        | <i>Boolean</i> | <p>Releases a user lock or group lock on the element object</p> <p>Throws an exception if the operation fails; use <i>GetLastError()</i> to retrieve error information</p>                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                      |
| <b>SaveLinkedDocument</b>                        | <i>Boolean</i> | Saves the linked document for this element to                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                      |

| Method                                                   | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See Also                                                       |
|----------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>ent (string Filename)</b>                             |                | the specified RTF file<br>Parameter:<br><ul style="list-style-type: none"> <li>• FileName: String - the name of the RTF file to which to save the linked document</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                |
| <b>SetAppearance (long Scope, long Item, long Value)</b> | <i>Void</i>    | Sets the visual appearance of the element<br>Parameter:<br><ul style="list-style-type: none"> <li>• Scope: Long - Scope of appearance set to modify<br/> <b>1</b> – Base (Default appearance across entire model)<br/><br/> To set appearance for the element (diagram object) in a selected diagram only, see <i>Setting The Style</i> in the <i>Diagram Object Class</i> topic</li> <li>• Item: Long - Appearance feature to modify<br/> <b>0</b> – Background color<br/> <b>1</b> – Font Color<br/> <b>2</b> – Border Color<br/> <b>3</b> – Border Width</li> <li>• Value: Long - Value to set appearance to</li> </ul> | <a href="#">DiagramObject Class</a> <small>[1926]</small>      |
| <b>UnlinkFromAssociation</b>                             | <i>Boolean</i> | Performs the opposite of <b>CreateAssociationClass()</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <a href="#">CreateAssociationClass()</a> <small>[1885]</small> |
| <b>Update ()</b>                                         | <i>Boolean</i> | Update the current element object after modification or appending a new item<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                |

#### 18.2.2.5.4 File Class

##### public Class

A *File* represents an associated file for an element. It is accessed through the Element *Files* collection.

Associated table in .EAP file: *t\_objectfiles*

##### File Attributes

| Attribute         | Type                                             | Notes                                                                                                    |
|-------------------|--------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>FileDate</b>   | <i>String</i>                                    | Read/Write. The file date when entry is created.                                                         |
| <b>Name</b>       | <i>String</i>                                    | Read/Write. The file name can be a logical file or a reference to a web address (using <i>http://</i> ). |
| <b>Notes</b>      | <i>String</i>                                    | Read/Write. Notes about the file.                                                                        |
| <b>ObjectType</b> | <a href="#">ObjectType</a> <small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                |

| Attribute | Type   | Notes                      |
|-----------|--------|----------------------------|
| Size      | String | Read/Write. The file size. |
| Type      | String | Read/Write. File type.     |

#### File Methods

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Update the current File object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                  |

#### 18.2.2.5.5 Issue (Maintenance) Class

An *Issue* is either a *Change* or a *Defect*, is associated with the containing element, and is accessed through the *Issues* collection of an element.

Associated table in .EAP file: *t\_objectproblems*

#### Issue Attributes

| Attribute     | Type                                                | Notes                                                                                                                         |
|---------------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| DateReported  | Date                                                | Read/Write. Date issue reported.                                                                                              |
| DateResolved  | Date                                                | Read/Write. Date issue resolved.                                                                                              |
| ElementID     | Long                                                | Read/Write. ID of element associated with this issue.                                                                         |
| Name          | String                                              | Read/Write. The Issue name; that is, the Issue itself.                                                                        |
| Notes         | String                                              | Read/Write. Issue description.                                                                                                |
| ObjectType    | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                     |
| Priority      | String                                              | Read/Write. Issue priority. Generally should use <b>Low</b> , <b>Medium</b> and <b>High</b> .                                 |
| Reporter      | String                                              | Read/Write. Person reporting issue.                                                                                           |
| Resolver      | String                                              | Read/Write. Person resolving issue.                                                                                           |
| ResolverNotes | String                                              | Read/Write. Notes entered by resolver about resolution.                                                                       |
| Severity      | String                                              | Read/Write. Issue severity. Should be marked as <b>Low</b> , <b>Medium</b> or <b>High</b> .                                   |
| Status        | String                                              | Read/Write. The current status of the issue.                                                                                  |
| Type          | Variant                                             | Read/Write. Issue type - can be <b>Defect</b> or <b>Change</b> , <b>Issue</b> and <b>ToDo</b> .                               |
| Version       | String                                              | Read/Write. Version associated with issue. Note that this method is only available through a Dispatch interface. For example: |

| Attribute | Type | Notes                                   |
|-----------|------|-----------------------------------------|
|           |      | Object ob = Issue;<br>Print ob.Version; |

#### Issue Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Issue object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                 |

#### 18.2.2.5.6 Metric Class

A *Metric* is a named item with a weighting that can be associated with an element for purposes of building metrics about the model. Accessed through the Element *Metrics* collection.

Associated table in .EAP file: *t\_objectmetrics*

#### Metric Attributes

| Attribute         | Type                                                | Notes                                                                     |
|-------------------|-----------------------------------------------------|---------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                                       | Read/Write. The name of the metric.                                       |
| <b>Notes</b>      | <i>String</i>                                       | Read/Write. Notes about this metric.                                      |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                                       | Read/Write. The metric type.                                              |
| <b>Weight</b>     | <i>Long</i>                                         | Read/Write. A user defined weighting for estimation or metric purposes.   |

#### Metric Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current Metric object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                |

### 18.2.2.5.7 Requirement Class

An *Element Requirement* object holds information about the responsibilities of an element in the context of the model. Accessed using the *Element Requirements* collection.

Associated table in .EAP file: *t\_objectrequires*

#### Requirement Attributes

| Attribute     | Type                                 | Notes                                                                                                                                                                                                                                            |
|---------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Difficulty    | String                               | Read/Write. Estimated difficulty to implement.                                                                                                                                                                                                   |
| LastUpdate    | Date                                 | Read/Write. Date requirement last updated.                                                                                                                                                                                                       |
| Name          | String                               | Read/Write. The requirement itself.                                                                                                                                                                                                              |
| Notes         | String                               | Read/Write. Further notes about requirement.                                                                                                                                                                                                     |
| ObjectType    | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                        |
| ParentID      | Long                                 | Read only. The <i>ElementID</i> of the element to which this requirement applies.                                                                                                                                                                |
| Priority      | String                               | Read/Write. Assigned priority of the requirement.                                                                                                                                                                                                |
| RequirementID | Long                                 | Read only. A local ID for this requirement.                                                                                                                                                                                                      |
| Stability     | String                               | Read/Write. Estimated stability of the requirement.<br><br>This is an indication of the probability of the requirement - or understanding of the requirement - changing. High stability indicates a low probability of the requirement changing. |
| Status        | String                               | Read/Write. Current status of the requirement.                                                                                                                                                                                                   |
| Type          | String                               | Read/Write. Requirement type.                                                                                                                                                                                                                    |

#### Requirement Methods

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Update the current Requirement object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.           |

### 18.2.2.5.8 Resource Class

An Element *Resource* is a named person/task pair with timing constraints and percent complete indicators. Use this to manage the work associated with delivering an Element.

Associated table in .EAP file: *t\_objectresources*

#### Resource Attributes

| Attribute       | Type                                 | Notes                                                                                  |
|-----------------|--------------------------------------|----------------------------------------------------------------------------------------|
| ActualHours     | Long                                 | Read/Write. Time already expended on the task, in hours, days or other units.          |
| DateEnd         | Date                                 | Read/Write. Expected end date.                                                         |
| DateStart       | Date                                 | Read/Write. Date to start work.                                                        |
| ExpectedHours   | Long                                 | Read/Write. The total expected time the task might run, in hours, days or other units. |
| History         | String                               | Read/Write. Gets or sets history text.                                                 |
| Name            | String                               | Read/Write. Name of resource (for example, person's name).                             |
| Notes           | String                               | Read/Write. Descriptive notes.                                                         |
| ObjectType      | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.              |
| PercentComplete | Long                                 | Read/Write. Current percent complete figure.                                           |
| Role            | String                               | Read/Write. Role they play in implementing the element.                                |
| Time            | Long                                 | Read/Write. Time expected; numeric indicating number of days.                          |

#### Resource Methods

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Update the current Resource object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.              |

### 18.2.2.5.9 Risk Class

A *Risk* object represents a named risk associated with an element and is used for project management purposes. Accessed through the Element *Risks* collection.

Associated table in .EAP file: *t\_objectrisks*

Risk Attributes

| Attribute  | Type                                                | Notes                                                                     |
|------------|-----------------------------------------------------|---------------------------------------------------------------------------|
| Name       | String                                              | Read/Write. The risk.                                                     |
| Notes      | String                                              | Read/Write. Further notes describing the risk.                            |
| ObjectType | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface. |
| Type       | String                                              | Read/Write. The risk type associated with this element.                   |
| Weight     | Long                                                | Read/Write. A weighting for estimation or metric purposes.                |

Risk Methods

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Update the current Risk object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                  |

**18.2.2.5.10 Scenario Class**

A *Scenario* corresponds to a Collaboration or Use Case instance. Each Scenario is a path of execution through the logic of a Use Case. Scenarios can be added to using the Element *Scenarios* collection.

Associated table in .EAP file: *t\_objectscenarios*

Scenario Attributes

| Attribute    | Type                                                                | Notes                                                                                                                                                                                                   | See Also                                                  |
|--------------|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| Name         | String                                                              | Read/Write. The Scenario name.                                                                                                                                                                          |                                                           |
| Notes        | String                                                              | Read/Write. Description of the Scenario. Usually contains the steps to execute the scenario.                                                                                                            |                                                           |
| ObjectType   | <a href="#">ObjectType</a><br><small>[1835]</small>                 | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                               |                                                           |
| ScenarioGUID | String                                                              | Read/Write. A unique ID for the Scenario. Used to identify the Scenario unambiguously within a model.                                                                                                   |                                                           |
| Steps        | Collection of <a href="#">ScenarioStep</a><br><small>[1894]</small> | Read only. A collection of step objects for this Scenario. Use the <b>AddNew</b> and <i>Delete</i> functions to manage steps. <i>AddNew</i> passes the step name and "1" as the type for an actor step. | <a href="#">Collection Class</a><br><small>[1841]</small> |
| Type         | String                                                              | Read/Write. The scenario type (for example, <i>Basic Path</i> ).                                                                                                                                        |                                                           |

| Attribute  | Type   | Notes                                                                                                                                                                    | See Also |
|------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Weight     | Long   | Read/Write. Currently used to position scenarios in the scenario list (that is, <i>List Position</i> ).                                                                  |          |
| XMLContent | String | Read/Write. A structured field that can contain scenario details in XML format. It is recommended that you use the <b>Steps</b> collection to read or modify this field. |          |

### Scenario Methods

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Update the current Scenario object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.              |

#### 18.2.2.5.11 ScenarioExtension Class

### ScenarioExtension Attributes

| Attribute     | Type                                                  | Notes                                                                                                                                                      |
|---------------|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ExtensionGUID | String                                                | Read/Write. A unique GUID for this Extension.                                                                                                              |
| Join          | String                                                | Read/Write. The GUID of the step where this Extension rejoins the Scenario.                                                                                |
| JoiningStep   | <a href="#">ScenarioStep</a><br><small>[1894]</small> | Read only. The actual step where this Extension rejoins the Scenario, if any.                                                                              |
| Level         | String                                                | Read only. The number of this Extension as shown in the scenario editor. This is derived from the value of <i>Pos</i> for this object and the owning step. |
| Name          | String                                                | Read/Write. The Extension name.<br><br><ul style="list-style-type: none"> <li>This should match the name of the linked scenario</li> </ul>                 |
| ObjectType    | <a href="#">ObjectType</a><br><small>[1835]</small>   | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                  |
| Pos           | Long                                                  | Read/Write. The position of the Extension in the Extensions list                                                                                           |
| Scenario      | <a href="#">Scenario</a><br><small>[1892]</small>     | Read only. The scenario that is executed as an alternative path for this Extension.                                                                        |

### ScenarioExtension Methods

| Method                 | Type           | Notes                                                                                                                                                                                    |
|------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.    |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>ScenarioExtension</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### 18.2.2.5.12 ScenarioStep Class

#### ScenarioStep Attributes

| Attribute              | Type                                   | Notes                                                                                                                                                                                                                            | See Also                                                                                                                                                    |
|------------------------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Extensions</b>      | <i>Collection of ScenarioExtension</i> | Read only. A collection of <i>ScenarioExtension</i> objects that specify how the scenario is extended from this step. The arguments to <b>AddNew</b> should match the name and GUID of the alternative scenario being linked to. | <a href="#">Collection</a> <sup>[1841]</sup><br><a href="#">ScenarioExtension</a> <sup>[1893]</sup>                                                         |
| <b>Level</b>           | <i>String</i>                          | Read only. The number of this Step as shown in the scenario editor. This is derived from the value of <i>Pos</i> .                                                                                                               |                                                                                                                                                             |
| <b>Link</b>            | <i>String</i>                          | Read/Write. The GUID of a Use Case that is relevant to this step.                                                                                                                                                                |                                                                                                                                                             |
| <b>LinkedElement</b>   | <i>Element</i>                         | Read only. The actual element specified by Link, if any.                                                                                                                                                                         | <a href="#">Element</a> <sup>[1877]</sup>                                                                                                                   |
| <b>Name</b>            | <i>String</i>                          | Read/Write. The Step name.                                                                                                                                                                                                       |                                                                                                                                                             |
| <b>ObjectType</b>      | <i>ObjectType</i>                      | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                        | <a href="#">ObjectType</a> <sup>[1835]</sup>                                                                                                                |
| <b>Pos</b>             | <i>Long</i>                            | Read/Write. The position of the Step in the Scenario Step list.                                                                                                                                                                  |                                                                                                                                                             |
| <b>Results</b>         | <i>String</i>                          | Read/Write. Any results that are given from this step.                                                                                                                                                                           |                                                                                                                                                             |
| <b>State</b>           | <i>String</i>                          | Read/Write. A description of the state the system enters when this Step is executed.                                                                                                                                             |                                                                                                                                                             |
| <b>StepGUID</b>        | <i>String</i>                          | Read/Write. A unique GUID for this Step.                                                                                                                                                                                         |                                                                                                                                                             |
| <b>StepType</b>        | <i>ScenarioStepType</i>                | Read/Write. Identifies whether this step is being performed by a user or the system.                                                                                                                                             | <a href="#">ScenarioStepType</a> <sup>[1838]</sup>                                                                                                          |
| <b>Uses</b>            | <i>String</i>                          | Read/Write. Input and requirements that are relevant to this step.                                                                                                                                                               |                                                                                                                                                             |
| <b>UsesElementList</b> | <i>Collection of Element</i>           | Read only. Indicates that the Structured Specification tab <b>Uses</b> field is a linked element list.                                                                                                                           | <a href="#">Collection</a> <sup>[1841]</sup><br><a href="#">Element</a> <sup>[1877]</sup><br><a href="#">Set Up Scenario Specification</a> <sup>[669]</sup> |

ScenarioStep Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>ScenarioStep</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.   |

**18.2.2.5.13 TaggedValue Class**

A *TaggedValue* is a named property and value associated with an element. It is accessed through the *TaggedValues* collection.

Associated table in .EAP file: *t\_objectproperties*

TaggedValue Attributes

| Attribute           | Type                                              | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ElementID</b>    | <i>Long</i>                                       | Read/Write. The local ID of the associated element.                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>FQName</b>       | <i>String</i>                                     | Read only. The fully-qualified name of the tag.                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Name</b>         | <i>String</i>                                     | Read/Write. Name of the tag.                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Notes</b>        | <i>String</i>                                     | Read/Write. Further descriptive notes about this tag.<br><br>If <b>Value</b> (below) is set to " <b>&lt;memo&gt;</b> ", then <b>Notes</b> should contain the actual Tagged Value content.                                                                                                                                                                                                                                            |
| <b>ObjectType</b>   | <a href="#">ObjectType</a><br><small>1835</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                                                                                                                                                                            |
| <b>PropertyGUID</b> | <i>String</i>                                     | Read/Write. The tag global ID.                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>PropertyID</b>   | <i>Long</i>                                       | Read only. The tag local ID.                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Value</b>        | <i>String</i>                                     | Read/Write. The value assigned to this tag.<br><br>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to " <b>&lt;memo&gt;</b> " and insert the body of text in the <b>Notes</b> attribute (above).<br><br>When reading existing Tagged Values, if <b>Value</b> = " <b>&lt;memo&gt;</b> " then the developer should read the actual body of text from the <b>Notes</b> attribute. |

TaggedValue Methods

| Method                | Type          | Notes                                                                    |
|-----------------------|---------------|--------------------------------------------------------------------------|
| <b>GetLastError()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in |

| Method          | Type           | Notes                                                                                                                                                                       |
|-----------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |                | relation to this object.<br>This function is rarely used as an exception is thrown when an error occurs.                                                                    |
| <b>Update()</b> | <i>Boolean</i> | Update the current TaggedValue object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 18.2.2.5.14 Test Class

A *Test* is a single Test Case applied to an element. Tests are added and accessed through the Element *Tests* collection.

Associated table in .EAP file: *t\_objecttests*

#### Test Attributes

| Attribute                 | Type                                 | Notes                                                                                                                                |
|---------------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>AcceptanceCriteria</b> | <i>String</i>                        | Read/Write. The acceptance criteria for successful execution.                                                                        |
| <b>CheckedBy</b>          | <i>String</i>                        | Read/Write. Results confirmed by.                                                                                                    |
| <b>Class</b>              | <i>Long</i>                          | Read/Write. The test Class:<br>1 = Unit Test<br>2 = Integration Test<br>3 = System Test<br>4 = Acceptance Test<br>5 = Scenario Test. |
| <b>DateRun</b>            | <i>Date</i>                          | Read/Write. Date last run.                                                                                                           |
| <b>Input</b>              | <i>String</i>                        | Read/Write. Input data.                                                                                                              |
| <b>Name</b>               | <i>String</i>                        | Read/Write. The test name.                                                                                                           |
| <b>Notes</b>              | <i>String</i>                        | Read/Write. Detailed notes about test to be carried out.                                                                             |
| <b>ObjectType</b>         | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.                                                            |
| <b>RunBy</b>              | <i>String</i>                        | Read/Write. Person conducting test.                                                                                                  |
| <b>Status</b>             | <i>String</i>                        | Read/Write. Current status of test.                                                                                                  |
| <b>TestResults</b>        | <i>Variant</i>                       | Read/Write. Results of test.                                                                                                         |
| <b>Type</b>               | <i>String</i>                        | Read/Write. The test type, such as Load or Regression.                                                                               |

#### Test Methods

| Method                 | Type          | Notes                                                                 |
|------------------------|---------------|-----------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred |

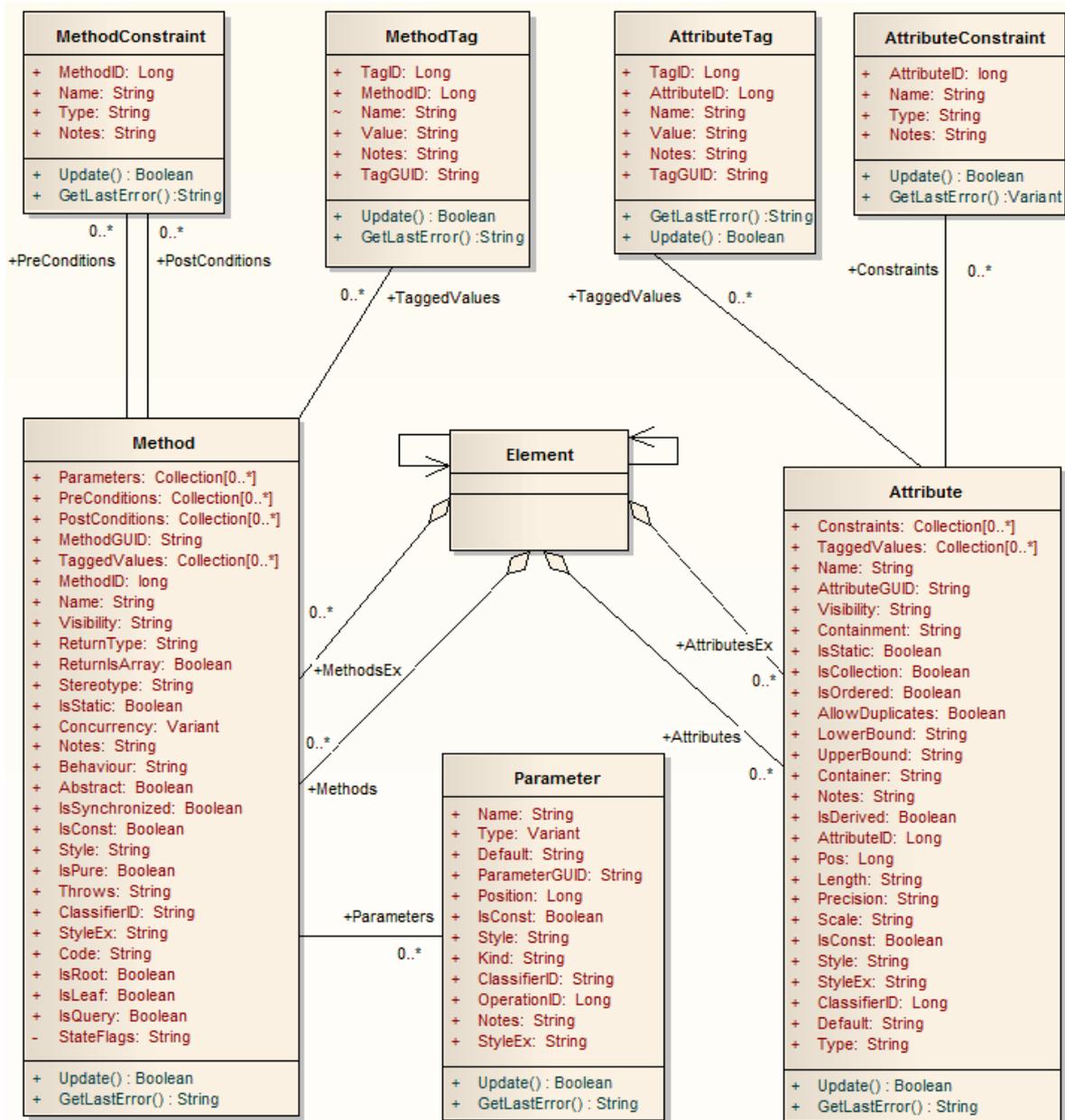
| Method           | Type           | Notes                                                                                                                                                                |
|------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  |                | in relation to this object.<br>This function is rarely used as an exception is thrown when an error occurs.                                                          |
| <b>Update ()</b> | <i>Boolean</i> | Update the current Test object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### 18.2.2.6 Element Features Package

The *ElementFeatures* package contains descriptions of the model interfaces that enable access to operations and attributes, and their associated Tagged Values and constraints.

This diagram illustrates the components associated with element features. These include *Attributes* and *Methods*, and the associated constraints and Tagged Values related to them. It also includes the *Parameter* object that defines the arguments associated with an operation (method).

#### Example:



**18.2.2.6.1 Attribute Class**

An *attribute* corresponds to a UML Attribute. It contains further collections for constraints and Tagged Values. Attributes are accessed from the Element *Attributes* collection.

Associated table in .EAP file: *t\_attribute*

**Attribute Attributes**

| Attribute              | Type           | Notes                                                                                                                                                                  |
|------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AllowDuplicates</b> | <i>Boolean</i> | Read/Write. Indicates if duplicates are allowed in the collection. If the attribute represents a database column, this when set represents the <b>Not Null</b> option. |

| Attribute     | Type                                 | Notes                                                                                                                                                                        |
|---------------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AttributeGUID | String                               | Read only. A globally unique ID for the current attribute. System generated.                                                                                                 |
| AttributeID   | Long                                 | Read only. Local ID number of the attribute.                                                                                                                                 |
| ClassifierID  | Long                                 | Read/Write. Classifier ID, if appropriate; indicates the base type associated with attribute, if not a primitive type.                                                       |
| Container     | String                               | Read/Write. The container type.                                                                                                                                              |
| Containment   | String                               | Read/Write. Type of containment. Can be <b>Not Specified, By Reference</b> or <b>By Value</b> .                                                                              |
| Constraints   | <a href="#">Collection</a><br>[1847] | Read only. A collection of <i>AttributeConstraint</i> objects. Used to access and manage constraints associated with this attribute.                                         |
| Default       | String                               | Read/Write. Initial value assigned to this attribute.                                                                                                                        |
| IsCollection  | Boolean                              | Read/Write. Indicates if the current feature is a collection or not. If the attribute represents a database column, this when set represents a Foreign Key.                  |
| IsConst       | Boolean                              | Read/Write. Flag indicating if the attribute is <b>Const</b> or not.                                                                                                         |
| IsDerived     | Boolean                              | Read/Write. Indicates if the attribute is derived (that is, a calculated value).                                                                                             |
| IsOrdered     | Boolean                              | Read/Write. Indicates if a collection is ordered or not. If the attribute represents a database column, this when set represents a Primary Key.                              |
| IsStatic      | Boolean                              | Read/Write. Indicates if the current attribute is a static feature or not. If the attribute represents a database column, this when set represents the <b>Unique</b> option. |
| Length        | String                               | Read/Write. The attribute length, where applicable.                                                                                                                          |
| LowerBound    | String                               | Read/Write. A value for the collection lower bound.                                                                                                                          |
| Name          | String                               | Read/Write. The attribute name.                                                                                                                                              |
| Notes         | String                               | Read/Write. Further notes about this attribute.                                                                                                                              |
| ObjectType    | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                    |
| ParentID      | Long                                 | Read only. Returns the <i>ElementID</i> of the element that this attribute is a part of.                                                                                     |
| Pos           | Long                                 | Read/Write. Position of the attribute in the Class attribute list.                                                                                                           |
| Precision     | String                               | Read/Write. Precision value.                                                                                                                                                 |
| Scale         | String                               | Read/Write. Scale value.                                                                                                                                                     |
| Stereotype    | String                               | Read/Write. Sets or gets the stereotype for this attribute.                                                                                                                  |
| StereotypeEx  | String                               | Read/Write. All the applied stereotypes of the attribute in a comma-separated list.                                                                                          |
| Style         | String                               | Read/Write. Contains the <b>Alias</b> property for this attribute.                                                                                                           |
| StyleEx       | String                               | Read/Write. Advanced style settings. Reserved for the use of Sparx Systems.                                                                                                  |

| Attribute             | Type                                                                                   | Notes                                                                                                                              |
|-----------------------|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>TaggedValues</b>   | <a href="#">Collection</a><br>[1847] of type<br><a href="#">AttributeTag</a><br>[1907] | Read only. A collection of AttributeTag objects. Use to access and manage Tagged Values associated with this attribute.            |
| <b>TaggedValuesEx</b> | <a href="#">Collection</a><br>[1847] of type<br><a href="#">TaggedValue</a><br>[1895]  | Read only. Collection of TaggedValue objects belonging to the current attribute and the TaggedValuesEx property of its classifier. |
| <b>Type</b>           | String                                                                                 | Read/Write. The attribute type (by name; also see <i>ClassifierID</i> ).                                                           |
| <b>UpperBound</b>     | String                                                                                 | Read/Write. A value for the collection upper bound.                                                                                |
| <b>Visibility</b>     | String                                                                                 | Read/Write. The scope of the attribute. Can be <b>Private</b> , <b>Protected</b> , <b>Public</b> or <b>Package</b> .               |

#### Attribute Methods

| Method                 | Type    | Notes                                                                                                                                                                                 |
|------------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | Boolean | Updates the current attribute object after modifying or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.               |

#### Notes:

- The Attribute Class in Enterprise Architect does not have an **Alias** property. You can access the **Alias** property of the Attribute Class using the **Style** property.

#### 18.2.2.6.2 **AttributeConstraint Class**

An *AttributeConstraint* is a constraint associated with the current Attribute.

Associated table in .EAP file: *t\_attributeconstraints*

#### AttributeConstraint Attributes

| Attribute          | Type                              | Notes                                                                     |
|--------------------|-----------------------------------|---------------------------------------------------------------------------|
| <b>AttributeID</b> | Long                              | Read/Write. ID of the attribute this constraint applies to.               |
| <b>Name</b>        | String                            | Read/Write. The constraint.                                               |
| <b>Notes</b>       | String                            | Read/Write. Descriptive notes about constraint.                           |
| <b>ObjectType</b>  | <a href="#">ObjectType</a> [1835] | Read only. Distinguishes objects referenced through a Dispatch interface. |

| Attribute | Type   | Notes                           |
|-----------|--------|---------------------------------|
| Type      | String | Read/Write. Type of constraint. |

#### AttributeConstraint Methods

| Method          | Type    | Notes                                                                                                                                                                                      |
|-----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.      |
| Update ()       | Boolean | Update the current <i>AttributeConstraint</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 18.2.2.6.3 AttributeTag Class

An *AttributeTag* represents a Tagged Value associated with an attribute.

Associated table in .EAP file: *t\_attributetag*

#### AttributeTag Attributes

| Attribute   | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AttributeID | Long                                 | Read/Write. Local ID of attribute associated with this Tagged Value.                                                                                                                                                                                                                                                                                                                                                                 |
| FQName      | String                               | Read only. The fully-qualified name of the tag.                                                                                                                                                                                                                                                                                                                                                                                      |
| Name        | String                               | Read/Write. Name of tag.                                                                                                                                                                                                                                                                                                                                                                                                             |
| Notes       | String                               | Read/Write. Further descriptive notes about this tag.<br><br>If <b>Value</b> (below) is set to " <b>&lt;memo&gt;</b> ", then <b>Notes</b> should contain the actual Tagged Value content.                                                                                                                                                                                                                                            |
| ObjectType  | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                                                                                                                                                                            |
| TagGUID     | String                               | Read/Write. A globally unique ID for this Tagged Value.                                                                                                                                                                                                                                                                                                                                                                              |
| TagID       | Long                                 | Read only. Local ID to identify Tagged Value.                                                                                                                                                                                                                                                                                                                                                                                        |
| Value       | String                               | Read/Write. The value assigned to this tag.<br><br>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to " <b>&lt;memo&gt;</b> " and insert the body of text in the <b>Notes</b> attribute (above).<br><br>When reading existing Tagged Values, if <b>Value</b> = " <b>&lt;memo&gt;</b> " then the developer should read the actual body of text from the <b>Notes</b> attribute. |

#### AttributeTag Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current <i>AttributeTag</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.  |

#### 18.2.2.6.4 CustomProperties Collection

The *CustomProperties* collection contains 0 or more *Cust Properties* associated with the current element. These properties provide advanced UML configuration options, and must not be added to or deleted. The value of each property can be set.

##### CustomProperty

| Attribute         | Type                                                | Notes                                                                                                                                                                                                                                                                    |
|-------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                                       | Read only. The CustomProperty name.                                                                                                                                                                                                                                      |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                |
| <b>Value</b>      | <i>String</i>                                       | Read/Write. The value associated with this custom property. Can be a string, the boolean values <b>true</b> or <b>false</b> , or an enumeration value from a defined list. The UML 2.3 specification in general provides information on enumeration kinds relevant here. |

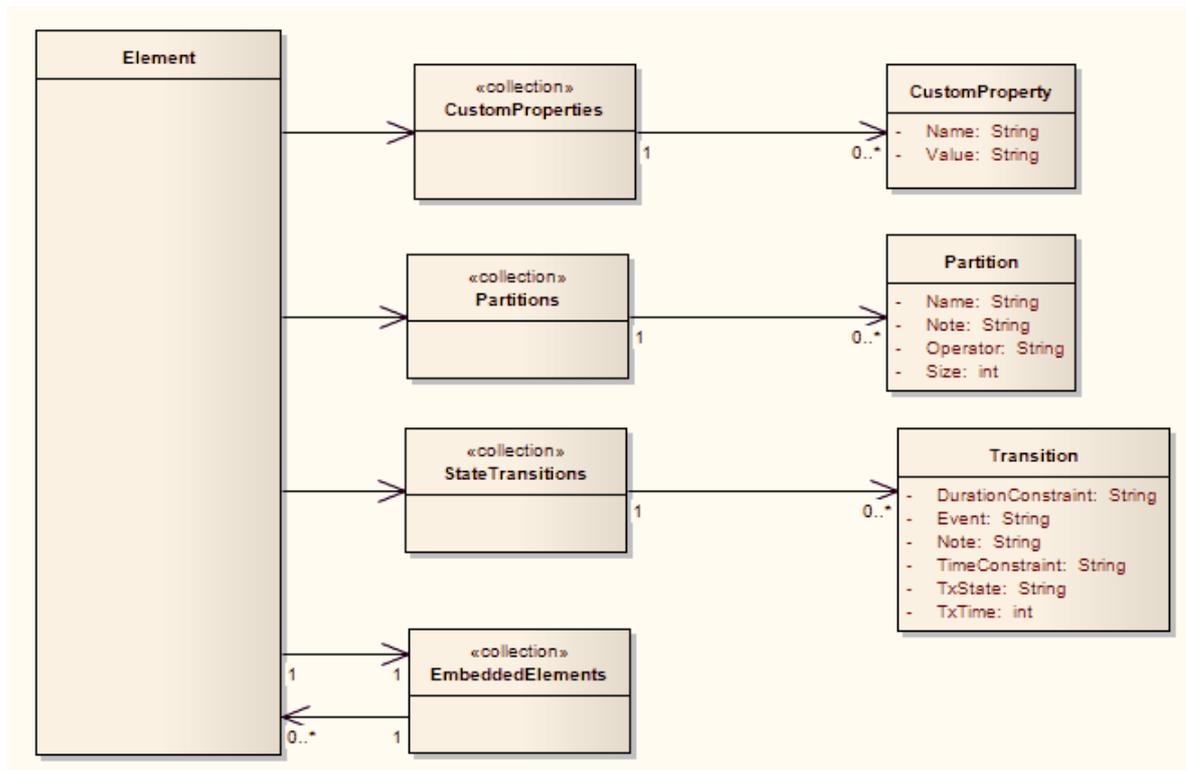
- The number and type of properties vary depending on the actual element

#### 18.2.2.6.5 EmbeddedElements Collection

In UML 2.3 an element can have one or more embedded elements such as Ports, Pins, Parameters or ObjectNodes. These are attached to the boundary of the host element and cannot be moved off the element. They are owned by their host element. This collection gives easy access to the set of elements embedded on the surface of an element. Note that some embedded elements can have their own embedded element collection (for example, Ports can have Interfaces embedded on them).

The *EmbeddedElements* collection contains Element objects.

##### Example:



#### 18.2.2.6.6 Method Class

A *method* represents a UML *operation*. It is accessed from the Element *Methods* collection and includes collections for parameters, constraints and Tagged Values.

Associated table in .EAP file: *t\_operation*

#### Method Attributes

| Attribute           | Type           | Notes                                                                                                                                                                                                                                                                                                         |
|---------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Abstract</b>     | <i>Boolean</i> | Read/Write. Flag indicating if the method is abstract ( <b>1</b> ) or not ( <b>0</b> ).                                                                                                                                                                                                                       |
| <b>Behavior</b>     | <i>String</i>  | Read/Write. Some further explanatory behavior notes (for example, pseudocode).<br><br>In earlier releases of Enterprise Architect this attribute had the UK/ Australian spelling 'Behaviour'; this is still present for backwards compatibility, but please now use the 'Behavior' attribute for consistency. |
| <b>ClassifierID</b> | <i>String</i>  | Read/Write. Classifier ID that applies to the <i>ReturnType</i> .                                                                                                                                                                                                                                             |
| <b>Code</b>         | <i>String</i>  | Read/Write. Optional field to hold the method Code (used for the <b>Initial Code</b> field).                                                                                                                                                                                                                  |
| <b>Concurrency</b>  | <i>Variant</i> | Read/Write. Concurrency type of method.                                                                                                                                                                                                                                                                       |
| <b>IsConst</b>      | <i>Boolean</i> | Read/Write. Flag indicating the method is <b>Const</b> .                                                                                                                                                                                                                                                      |
| <b>IsLeaf</b>       | <i>Boolean</i> | Read/Write. Flag to indicate if the method is <i>Leaf</i> (cannot be overridden).                                                                                                                                                                                                                             |
| <b>IsPure</b>       | <i>Boolean</i> | Read/Write. Flag indicating the method is defined as Pure in C++.                                                                                                                                                                                                                                             |

| Attribute      | Type                                                                                                                 | Notes                                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| IsQuery        | Boolean                                                                                                              | Read/Write. Flag to indicate if the method is a query (that is, does not alter Class variables).                                     |
| IsRoot         | Boolean                                                                                                              | Read/Write. Flag to indicate if the method is <i>Root</i> .                                                                          |
| IsStatic       | Boolean                                                                                                              | Read/Write. Flag to indicate a static method.                                                                                        |
| IsSynchronized | Boolean                                                                                                              | Read/Write. Flag indicating a Synchronized method call.                                                                              |
| MethodGUID     | String                                                                                                               | Read/Write. A globally unique ID for the current method. System generated.                                                           |
| MethodID       | Long                                                                                                                 | Read only. A local ID for the current method, only valid within this .EAP file.                                                      |
| Name           | String                                                                                                               | Read/Write. The method name.                                                                                                         |
| Notes          | String                                                                                                               | Read/Write. Descriptive notes about the method.                                                                                      |
| ObjectType     | <a href="#">ObjectType</a><br><small>[1835]</small>                                                                  | Read only. Distinguishes objects referenced through a Dispatch interface.                                                            |
| Parameters     | <a href="#">Collection</a><br><small>[1841]</small>                                                                  | Read only. The <i>Parameters</i> collection for the current method. Use to add and access parameter objects for the current method.  |
| ParentID       | Long                                                                                                                 | Read only. Returns the <i>ElementID</i> of the element that this method belongs to.                                                  |
| Pos            | Long                                                                                                                 | Read/Write. Specifies the position of the method within the set of operations defined for a Class.                                   |
| PostConditions | <a href="#">Collection</a><br><small>[1841]</small>                                                                  | Read only. PostConditions (constraints) as they apply to this method. Returns a <i>MethodConstraint</i> object of type <b>post</b> . |
| PreConditions  | <a href="#">Collection</a><br><small>[1841]</small>                                                                  | Read only. PreConditions (constraints) as they apply to this method. Returns a <i>MethodConstraint</i> object of type <b>pre</b> .   |
| ReturnsArray   | Boolean                                                                                                              | Read/Write. Flag to indicate the return value is an array.                                                                           |
| ReturnType     | String                                                                                                               | Read/Write. Return type for the method; can be a primitive data type or a Class or Interface type.                                   |
| StateFlags     | String                                                                                                               | Read/Write. Some flags as applied to methods in State elements.                                                                      |
| Stereotype     | String                                                                                                               | Read/Write. The method stereotype (optional).                                                                                        |
| StereotypeEx   | String                                                                                                               | Read/Write. All the applied stereotypes of the method in a comma-separated list.                                                     |
| Style          | String                                                                                                               | Read/Write. Contains the <b>Alias</b> property for this method.                                                                      |
| StyleEx        | String                                                                                                               | Read/Write. Advanced style settings. Reserved for the use of Sparx Systems.                                                          |
| TaggedValues   | <a href="#">Collection</a><br><small>[1841]</small><br>of type<br><a href="#">MethodTag</a><br><small>[1906]</small> | Read only. <i>TaggedValues</i> collection for the current method. Accesses a list of <i>MethodTag</i> objects.                       |
| Throws         | String                                                                                                               | Read/Write. Exception information.                                                                                                   |
| Visibility     | String                                                                                                               | Read/Write. The method scope: <b>Public</b> , <b>Protected</b> , <b>Private</b> or <b>Package</b> .                                  |

Method Methods

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current method object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                |

**18.2.2.6.7 MethodConstraint Class**

A *MethodConstraint* is a condition imposed on a method. It is accessed through either the Method *PreConditions* or Method *PostConditions* collection.

Associated table in .EAP file: *t\_operationpres* and *t\_operationposts*

MethodConstraint Attributes

| Attribute         | Type                                 | Notes                                                                     |
|-------------------|--------------------------------------|---------------------------------------------------------------------------|
| <b>MethodID</b>   | <i>Long</i>                          | Read/Write. The local ID of the associated method.                        |
| <b>Name</b>       | <i>String</i>                        | Read/Write. The name of the constraint.                                   |
| <b>Notes</b>      | <i>String</i>                        | Read/Write. Descriptive notes about this constraint.                      |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface. |
| <b>Type</b>       | <i>String</i>                        | Read/Write. The constraint type.                                          |

MethodConstraint Methods

| Method                 | Type           | Notes                                                                                                                                                                                   |
|------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.   |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>MethodConstraint</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### 18.2.2.6.8 MethodTag Class

A *MethodTag* is a Tagged Value associated with a method.

Associated table in .EAP file: *t\_operationtag*

#### MethodTag Attributes

| Attribute         | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FQName</b>     | <i>String</i>                                       | Read only. The fully-qualified name of the tag.                                                                                                                                                                                                                                                                                                                                                                              |
| <b>MethodID</b>   | <i>Long</i>                                         | Read/Write. The ID of the associated method.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Name</b>       | <i>String</i>                                       | Read/Write. The tag or name of the property.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Notes</b>      | <i>String</i>                                       | Read/Write. Further descriptive notes about this tag.<br>If <b>Value</b> (below) is set to " <b>&lt;memo&gt;</b> ", then <b>Notes</b> should contain the actual Tagged Value content.                                                                                                                                                                                                                                        |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                                                                                                                                                                    |
| <b>TagGUID</b>    | <i>String</i>                                       | Read/Write. A unique GUID for this Tagged Value.                                                                                                                                                                                                                                                                                                                                                                             |
| <b>TagID</b>      | <i>Long</i>                                         | Read only. A unique ID for this Tagged Value.                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Value</b>      | <i>String</i>                                       | Read/Write. The value assigned to this tag.<br>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to " <b>&lt;memo&gt;</b> " and insert the body of text in the <b>Notes</b> attribute (above).<br>When reading existing Tagged Values, if <b>Value</b> = " <b>&lt;memo&gt;</b> " then the developer should read the actual body of text from the <b>Notes</b> attribute. |

#### MethodTag Methods

| Method                 | Type           | Notes                                                                                                                                                                             |
|------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Updates the current <i>MethodTag</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### 18.2.2.6.9 Parameter Class

A *Parameter* object represents a method argument and is accessed through the *Method Parameters* collection.

Associated table in .EAP file: *t\_operationparams*

**Parameter Attributes**

| Attribute     | Type                                                                               | Notes                                                                                    |
|---------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Alias         | String                                                                             | Read/Write. An optional alias for this parameter.                                        |
| ClassifierID  | String                                                                             | Read/Write. A ClassifierID for the parameter, if known.                                  |
| Default       | String                                                                             | Read/Write. A default value for this parameter.                                          |
| IsConst       | Boolean                                                                            | Read/Write. Flag indicating the parameter is <i>Const</i> (cannot be altered).           |
| Kind          | String                                                                             | Read/Write. The parameter kind - <b>in</b> , <b>inout</b> , <b>out</b> , <b>return</b> . |
| Name          | String                                                                             | Read/Write. The parameter name; must be unique for a single method.                      |
| Notes         | String                                                                             | Read/Write. Descriptive notes.                                                           |
| ObjectType    | <a href="#">ObjectType</a><br>[1835]                                               | Read only. Distinguishes objects referenced through a Dispatch interface.                |
| OperationID   | Long                                                                               | Read only. ID of the method associated with this parameter.                              |
| ParameterGUID | String                                                                             | Read/Write. A globally unique ID for the current Parameter. System generated.            |
| Position      | Long                                                                               | Read/Write. The position in the argument list.                                           |
| Stereotype    | String                                                                             | Read/Write. The first stereotype of the parameter.                                       |
| StereotypeEx  | String                                                                             | Read/Write. All the applied stereotypes of the parameter in a comma-separated list.      |
| Style         | String                                                                             | Read/Write. Some style information.                                                      |
| StyleEx       | String                                                                             | Read/Write. Advanced style settings. Reserved for the use of Sparx Systems.              |
| TaggedValues  | <a href="#">Collection</a><br>[1847] of type<br><a href="#">ParamTag</a><br>[1908] | Read/Write. GUID of the parameter with which this <i>ParamTag</i> is associated.         |
| Type          | Variant                                                                            | Read/Write. The parameter type; can be a primitive type or defined classifier.           |

**Parameter Methods**

| Method          | Type    | Notes                                                                                                                                                                                 |
|-----------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| Update ()       | Boolean | Update the current Parameter object after modifying or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                |

### 18.2.2.6.10 ParamTag Class

A *ParamTag* is a Tagged Value associated with a method parameter.

#### ParamTag Attributes

| Attribute    | Type                                                | Notes                                                                     |
|--------------|-----------------------------------------------------|---------------------------------------------------------------------------|
| ElementGUID  | String                                              | Read/Write. GUID of the parameter with which this ParamTag is associated. |
| FQName       | String                                              | Read only. The fully qualified name of the tag.                           |
| ObjectType   | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface. |
| PropertyGUID | String                                              | Read/Write. A system generated GUID to identify the Tagged Value.         |
| Tag          | String                                              | Read/Write. The actual tag name.                                          |
| Value        | String                                              | Read/Write. The value associated with this tag.                           |

#### ParamTag Methods

| Method          | Type    | Notes                                                                                                                                                                         |
|-----------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.                                                                             |
| Update ()       | Boolean | Updates the current <i>ParamTag</i> object after modifying or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

### 18.2.2.6.11 Partitions Collection

A collection of internal element partitions (regions). This is commonly seen in **Activity, State, Boundary, Diagram Frame** and similar elements. Not all elements support partitions.

This collection contains a set of *Partition* elements. The set is read/write: information is not saved until the host element is saved, so ensure that you call the *Element.Save* method after making changes to a Partition.

#### Partition Attributes

| Attribute  | Type                                                | Notes                                                                                  |
|------------|-----------------------------------------------------|----------------------------------------------------------------------------------------|
| Name       | String                                              | Read/Write. The partition name; can represent a condition or constraint in some cases. |
| Note       | String                                              | Read/Write. A free text note associated with this partition.                           |
| ObjectType | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.              |
| Operator   | String                                              | Read/Write. An optional operator value that specifies the partition type.              |

| Attribute   | Type          | Notes                                                            |
|-------------|---------------|------------------------------------------------------------------|
| <b>Size</b> | <i>String</i> | Read/Write. Vertical or horizontal width of partition in pixels. |

**Learn More:**

- [Activity](#)<sup>[873]</sup>
- [State](#)<sup>[917]</sup>
- [Boundary](#)<sup>[1288]</sup>
- [Diagram Frame](#)<sup>[888]</sup>

**18.2.2.6.12 Properties Class****Properties**

Properties Attributes

| Attribute         | Type                                         | Notes                                                                     |
|-------------------|----------------------------------------------|---------------------------------------------------------------------------|
| <b>Count</b>      | <i>Long</i>                                  | The number of properties that are available for this object.              |
| <b>ObjectType</b> | <a href="#">ObjectType</a> <sup>[1835]</sup> | Read only. Distinguishes objects referenced through a Dispatch interface. |

Properties Methods

| Method                     | Type            | Notes                                                                                                                                                                                                                                                                                                         |
|----------------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Item (object Index)</b> | <i>Property</i> | Returns a property either by name or by zero-based integer offset into the list of properties.<br><br>Parameter: <ul style="list-style-type: none"> <li>• Index: Variant - either a string representing the property name or an integer representing the zero-based offset into the property list.</li> </ul> |

**Property**

Property Attributes

| Attribute         | Type                                         | Notes                                                                                                                                                                                                                                                       |
|-------------------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name</b>       | <i>String</i>                                | Read only. Identifies the property. The object to which the properties list applies can have an automation property with the same name, in which case the data accessed through <b>Value</b> is identical to that obtained through the automation property. |
| <b>ObjectType</b> | <a href="#">ObjectType</a> <sup>[1835]</sup> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                   |
| <b>Type</b>       | <a href="#">PropType</a> <sup>[1836]</sup>   | Read only. Provides an indication of what sort of data is going to be stored by this property. This restriction can be further defined by the Validation attribute.                                                                                         |

| Attribute  | Type    | Notes                                                                                                                                                                                                                                                                            |
|------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Validation | String  | Read only. Optional string that is used to validate any data that is passed to the Value attribute. This string is used by the programmer at run time to provide an indication of what's expected, and by Enterprise Architect to ensure that the submitted data is appropriate. |
| Value      | Variant | Read/write. The value of the property as defined in the other fields.                                                                                                                                                                                                            |

#### 18.2.2.6.13 Transitions Collection

Applies only to *Timeline elements*. A Timeline element displays 0 or more state transitions at set times on its extent. This collection enables you to access the transition set. You can also access additional information by referring to the connectors associated with the Timeline, and by referencing messages passed between timelines. Note that any changes made to elements in this collection are only saved when the main element is saved.

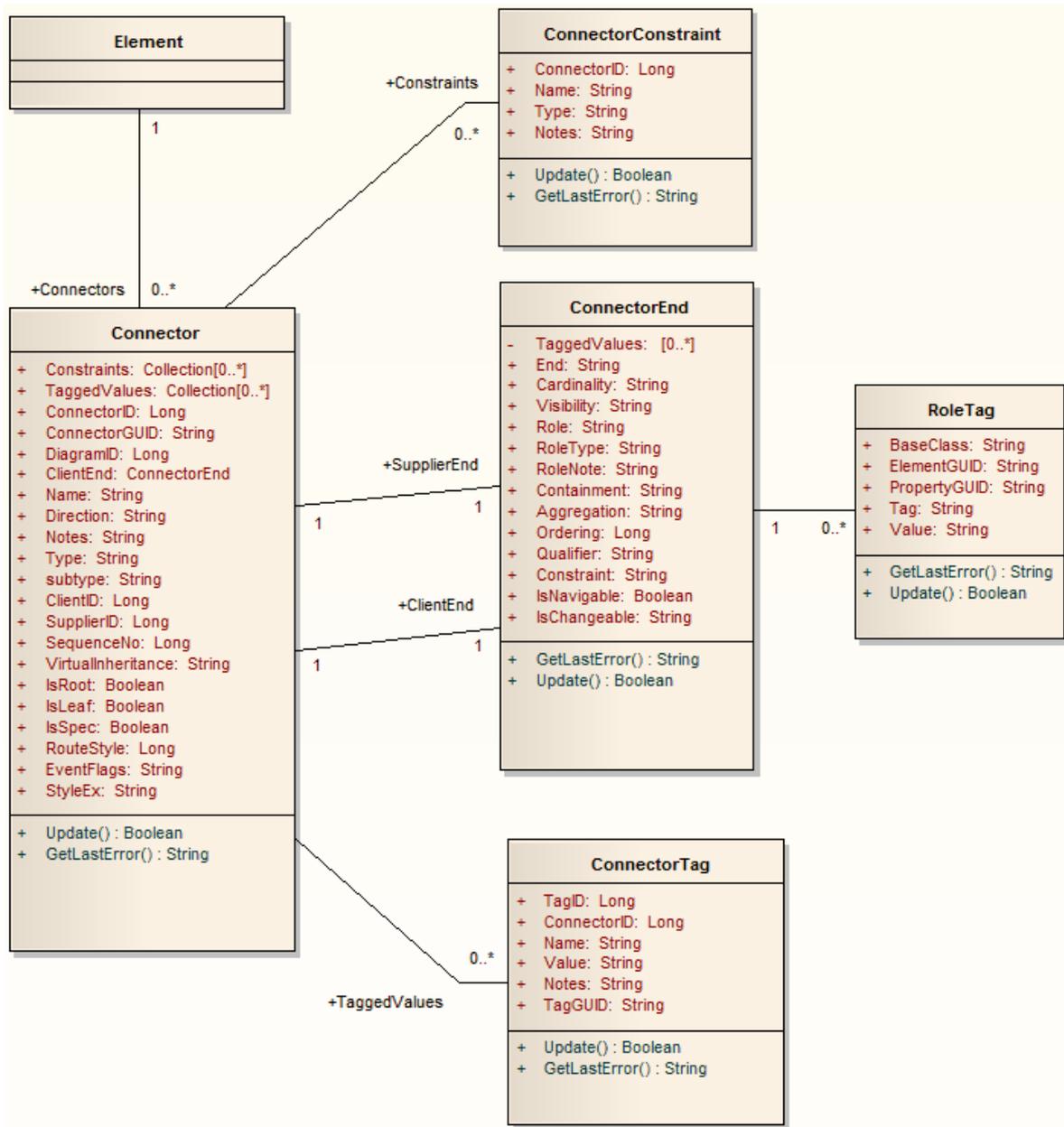
##### Transition Attributes

| Attribute          | Type                                 | Notes                                                                                     |
|--------------------|--------------------------------------|-------------------------------------------------------------------------------------------|
| DurationConstraint | String                               | Read/Write. A constraint on the time duration that the transition takes.                  |
| Event              | String                               | Read/Write. Event (optional) that initiated transition.                                   |
| Note               | String                               | Read/Write. A free text note.                                                             |
| ObjectType         | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.                 |
| TimeConstraint     | String                               | Read/Write. A constraint on when the transition has to be complete by.                    |
| TxState            | String                               | Read/Write. The state to transition to. Defined in the <b>Timeline Properties</b> dialog. |
| TxTime             | String                               | Read/Write. The time that the transition occurs. Value depends on range set in diagram.   |

#### 18.2.2.7 Connector Package

The *Connector* package details how connectors between elements are accessed and managed.

##### Example:



### 18.2.2.7.1 Connector Class

#### Reference:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                              | See also |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b> | <p>A <i>Connector</i> object represents the various kinds of connectors between UML elements; it is accessed from either the <i>Client</i> or <i>Supplier</i> element, using the <i>Connectors</i> collection of that element</p> <p>When creating a new connector you must assign it a valid type from the following list:</p> <ul style="list-style-type: none"> <li>• Aggregation</li> <li>• Assembly</li> </ul> |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <ul style="list-style-type: none"> <li>• Association</li> <li>• Collaboration</li> <li>• CommunicationPath</li> <li>• Connector</li> <li>• ControlFlow</li> <li>• Delegate</li> <li>• Dependency</li> <li>• Deployment</li> <li>• ERLink</li> <li>• Generalization</li> <li>• InformationFlow</li> <li>• Instantiation</li> <li>• InterruptFlow</li> <li>• Manifest</li> <li>• Nesting</li> <li>• NoteLink</li> <li>• ObjectFlow</li> <li>• Package</li> <li>• Realization</li> <li>• Sequence</li> <li>• StateFlow</li> <li>• UseCase</li> </ul> <p>Associated table in .EAP file: <i>t_connector</i></p> |          |

#### Connector Attributes

| Attribute            | Type                                           | Notes                                                                                                    | See Also                                     |
|----------------------|------------------------------------------------|----------------------------------------------------------------------------------------------------------|----------------------------------------------|
| <b>Alias</b>         | <i>String</i>                                  | Read/Write<br>An optional alias for this connector                                                       |                                              |
| <b>ClientEnd</b>     | <a href="#">ConnectorEnd</a> <sup>[1916]</sup> | Read only<br>A pointer to the <i>ConnectorEnd</i> object representing the source end of the relationship |                                              |
| <b>ClientID</b>      | <i>Long</i>                                    | Read/Write<br><i>ElementID</i> of the element at the source end of this connector                        |                                              |
| <b>Color</b>         | <i>Long</i>                                    | Read/Write<br>Sets the color of the connector                                                            |                                              |
| <b>ConnectorGUID</b> | <i>Variant</i>                                 | Read only<br>A globally unique ID for the current connector; system generated                            |                                              |
| <b>ConnectorID</b>   | <i>Long</i>                                    | Read only<br>Local identifier for the current connector; system generated                                |                                              |
| <b>Constraints</b>   | <a href="#">Collection</a> <sup>[1841]</sup>   | Read only                                                                                                | <a href="#">Constraint</a> <sup>[1876]</sup> |

| Attribute               | Type                                                | Notes                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also                                                           |
|-------------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
|                         |                                                     | Collection of <b>constraint</b> objects                                                                                                                                                                                                                                                                                                                                                                                      |                                                                    |
| <b>CustomProperties</b> | <a href="#">Collection</a><br><small>[1841]</small> | Read only<br>Returns a collection of advanced properties associated with an element in the form of <b>CustomProperty</b> objects                                                                                                                                                                                                                                                                                             | <a href="#">CustomProperty Collection</a><br><small>[1902]</small> |
| <b>DiagramID</b>        | <i>Long</i>                                         | Read/Write<br>The <i>DiagramID</i> of the connector                                                                                                                                                                                                                                                                                                                                                                          |                                                                    |
| <b>Direction</b>        | <i>String</i>                                       | Read/Write<br>Connector direction; can be set to one of the following: <ul style="list-style-type: none"> <li>• <b>Unspecified</b></li> <li>• <b>Bi-Directional</b></li> <li>• <b>Source -&gt; Destination</b></li> <li>• <b>Destination -&gt; Source</b></li> </ul>                                                                                                                                                         |                                                                    |
| <b>EndPointX</b>        | <i>Long</i>                                         | Read/Write<br>The x-coordinate of the connector's end point<br>Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen                                                                                                                                                                                                                                                     |                                                                    |
| <b>EndPointY</b>        | <i>Long</i>                                         | Read/Write<br>The y-coordinate of the connector's end point<br>Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen                                                                                                                                                                                                                                                     |                                                                    |
| <b>EventFlags</b>       | <i>String</i>                                       | Read/Write<br>Structure to hold a variety of flags concerned with event signaling on messages                                                                                                                                                                                                                                                                                                                                |                                                                    |
| <b>IsLeaf</b>           | <i>Boolean</i>                                      | Read/Write<br>Flag indicating connector is a <i>leaf</i>                                                                                                                                                                                                                                                                                                                                                                     |                                                                    |
| <b>IsRoot</b>           | <i>Boolean</i>                                      | Read/Write<br>Flag indicating connector is a <i>root</i>                                                                                                                                                                                                                                                                                                                                                                     |                                                                    |
| <b>IsSpec</b>           | <i>Boolean</i>                                      | Read/Write<br>Flag indicating connector is a specification                                                                                                                                                                                                                                                                                                                                                                   |                                                                    |
| <b>MetaType</b>         | <i>String</i>                                       | Read only<br>The connector's domain-specific meta type, as defined by an applied stereotype from an MDG Technology                                                                                                                                                                                                                                                                                                           |                                                                    |
| <b>MiscData</b>         | <i>String</i>                                       | Read only<br>This low-level property returns an array providing information about the contents of the <b>PData</b> fields; these database fields are not documented and developers must gain understanding of these fields through their own endeavors to use this property<br><b>MiscData</b> is zero based, therefore: <ul style="list-style-type: none"> <li>• <b>MiscData(0)</b> corresponds to <b>PData1</b></li> </ul> |                                                                    |

| Attribute           | Type                                                           | Notes                                                                                                                                                                                                                                                                                            | See Also |
|---------------------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                     |                                                                | <ul style="list-style-type: none"> <li>• <b>MiscData(1)</b> to <b>PData2</b></li> </ul> and so on                                                                                                                                                                                                |          |
| <b>Name</b>         | <i>String</i>                                                  | Read/Write<br>The connector name                                                                                                                                                                                                                                                                 |          |
| <b>Notes</b>        | <i>String</i>                                                  | Read/Write<br>Descriptive notes about the connector                                                                                                                                                                                                                                              |          |
| <b>ObjectType</b>   | <a href="#">ObjectTyp</a><br><small>e<sup>[1835]</sup></small> | Read only<br>Distinguishes objects referenced through a Dispatch interface                                                                                                                                                                                                                       |          |
| <b>Properties</b>   | <a href="#">Properties</a><br><small>[1909]</small>            | Returns a list of specialized properties applicable to the connector that might not be available using the automation model<br><br>The properties are purposely undocumented because of their obscure nature and because they are subject to change as progressive enhancements are made to them |          |
| <b>RouteStyle</b>   | <i>Long</i>                                                    | Read/Write<br>The route style                                                                                                                                                                                                                                                                    |          |
| <b>SequenceNo</b>   | <i>Long</i>                                                    | Read/Write<br>The <i>SequenceNo</i> of the connector                                                                                                                                                                                                                                             |          |
| <b>StartPointX</b>  | <i>Long</i>                                                    | Read/Write<br>The x-coordinate of the connector's start point<br>Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen                                                                                                                       |          |
| <b>StartPointY</b>  | <i>Long</i>                                                    | Read/Write<br>The y-coordinate of the connector's start point<br>Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen                                                                                                                       |          |
| <b>StateFlags</b>   | <i>String</i>                                                  | Read/Write<br>Structure to hold a variety of flags concerned with State signaling on messages, the list delimited by semi-colons                                                                                                                                                                 |          |
| <b>Stereotype</b>   | <i>String</i>                                                  | Read/Write<br>Sets or gets the stereotype for this connector end                                                                                                                                                                                                                                 |          |
| <b>StereotypeEx</b> | <i>String</i>                                                  | Read/Write<br>All the applied stereotypes of the connector in a comma-separated list                                                                                                                                                                                                             |          |
| <b>StyleEx</b>      | <i>String</i>                                                  | Read/Write<br>Advanced style settings; reserved for the use of Sparx Systems                                                                                                                                                                                                                     |          |
| <b>Subtype</b>      | <i>String</i>                                                  | Read/Write                                                                                                                                                                                                                                                                                       |          |

| Attribute                 | Type                                           | Notes                                                                                                    | See Also                                     |
|---------------------------|------------------------------------------------|----------------------------------------------------------------------------------------------------------|----------------------------------------------|
|                           |                                                | A possible subtype to refine the meaning of the connector                                                |                                              |
| <b>SupplierEnd</b>        | <a href="#">ConnectorEnd</a> <sup>[1916]</sup> | Read only<br>A pointer to the <i>ConnectorEnd</i> object representing the target end of the relationship |                                              |
| <b>SupplierID</b>         | <i>Long</i>                                    | Read/Write<br><i>ElementID</i> of the element at the target end of this connector                        |                                              |
| <b>TaggedValues</b>       | <i>Collection</i>                              | Read only<br>Collection of <i>ConnectorTag</i> objects                                                   |                                              |
| <b>TransitionAction</b>   | <i>String</i>                                  | Read/Write<br>See the <i>Transition</i> topic for appropriate values                                     | <a href="#">Transition</a> <sup>[1013]</sup> |
| <b>TransitionEvent</b>    | <i>String</i>                                  | Read/Write<br>See the <i>Transition</i> topic for appropriate values                                     | <a href="#">Transition</a> <sup>[1013]</sup> |
| <b>TransitionGuard</b>    | <i>String</i>                                  | Read/Write<br>See the <i>Transition</i> topic for appropriate values                                     | <a href="#">Transition</a> <sup>[1013]</sup> |
| <b>Type</b>               | <i>String</i>                                  | Read/Write<br>Connector type; valid types are held in the <i>t_connectortypes</i> table in the .EAP file |                                              |
| <b>VirtualInheritance</b> | <i>String</i>                                  | Read/Write<br>For <i>Generalization</i> , indicates if inheritance is virtual                            |                                              |
| <b>Width</b>              | <i>Long</i>                                    | Read/Write<br>Specifies the width of the connector                                                       |                                              |

### Connector Methods

| Method                 | Type           | Notes                                                                                                                                                                                |
|------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object<br><br>This function is rarely used, as an exception is thrown when an error occurs |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>ConnectorObject</i> after modification or appending a new item<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information |

#### 18.2.2.7.2 ConnectorConstraint Class

A *ConnectorConstraint* holds information about special conditions that apply to a connector. It is accessed through the Connector *Constraints* collection.

Associated table in .EAP file: *t\_connectorconstraints*

ConnectorConstraint Attributes

| Attribute   | Type                                 | Notes                                                                     |
|-------------|--------------------------------------|---------------------------------------------------------------------------|
| ConnectorID | Long                                 | Read/Write. A local ID value (long) - system generated.                   |
| Name        | String                               | Read/Write. The constraint name.                                          |
| Notes       | String                               | Read/Write. Notes about this constraint.                                  |
| ObjectType  | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface. |
| Type        | String                               | Read/Write. The constraint type.                                          |

ConnectorConstraint Methods

| Method          | Type    | Notes                                                                                                                                                                                      |
|-----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.      |
| Update ()       | Boolean | Update the current <i>ConnectorConstraint</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

**18.2.2.7.3 ConnectorEnd Class**

A *ConnectorEnd* contains information about a single end of a connector. A *ConnectorEnd* is accessed from the connector as either the *ClientEnd* or *SupplierEnd*.

Associated table in .EAP file: derived from *t\_connector*

ConnectorEnd Attributes

| Attribute           | Type    | Notes                                                                                                                | See Also |
|---------------------|---------|----------------------------------------------------------------------------------------------------------------------|----------|
| Aggregation         | Long    | Read/Write. Aggregation as it applies to this end. Valid values are:<br><br>0 = None<br>1 = Shared<br>2 = Composite. |          |
| Alias               | String  | Read/Write. An optional alias for this connector end.                                                                |          |
| AllowDuplicat<br>es | Boolean | Read/Write. For multiplicities greater than 1, indicates that duplicate entries are possible.                        |          |
| Cardinality         | String  | Read/Write. Cardinality associated with this end.                                                                    |          |
| Constraint          | String  | Read/Write. A constraint that can be applied to this connector end.                                                  |          |
| Containment         | String  | Read/Write. Containment type applied to this connector                                                               |          |

| Attribute                | Type                                                | Notes                                                                                                                                                                                         | See Also                                               |
|--------------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
|                          |                                                     | end.                                                                                                                                                                                          |                                                        |
| <b>Derived</b>           | <i>Boolean</i>                                      | Read/Write. Indicates that the value of this end is derived.                                                                                                                                  |                                                        |
| <b>DerivedUnion</b>      | <i>Boolean</i>                                      | Read/Write. Indicates the value of this role derived from the union of all roles that subset this.                                                                                            |                                                        |
| <b>End</b>               | <i>String</i>                                       | Read only. The end this <i>ConnectorEnd</i> object applies to: <i>Client</i> or <i>Supplier</i> .                                                                                             |                                                        |
| <b>IsChangeable</b>      | <i>String</i>                                       | Read/Write. Flag indicating whether this end is changeable or not.<br>Values: <b>frozen</b> , <b>addOnly</b> or <b>none</b> .                                                                 |                                                        |
| <b>IsNavigable</b>       | <i>Boolean</i>                                      | Read/Write. Flag indicating this end is navigable from the other end.                                                                                                                         |                                                        |
| <b>Navigable</b>         | <i>String</i>                                       | Read/Write. Indicates whether this role of an association is navigable from the opposite classifier. Three values are valid: <i>Navigable</i> , <i>Non-Navigable</i> and <i>Unspecified</i> . |                                                        |
| <b>ObjectType</b>        | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                     |                                                        |
| <b>Ordering</b>          | <i>Long</i>                                         | Read/Write. Ordering for this connector end.                                                                                                                                                  |                                                        |
| <b>OwnedByClassifier</b> | <i>Boolean</i>                                      | Read/Write. Indicates this Association end corresponds to an attribute on the opposite end of the Association.                                                                                |                                                        |
| <b>Qualifier</b>         | <i>String</i>                                       | Read/Write. A qualifier that can apply to connector end.                                                                                                                                      |                                                        |
| <b>Role</b>              | <i>String</i>                                       | Read/Write. The connector end role.                                                                                                                                                           |                                                        |
| <b>RoleNote</b>          | <i>String</i>                                       | Read/Write. Notes associated with the role of this connector end.                                                                                                                             |                                                        |
| <b>RoleType</b>          | <i>String</i>                                       | Read/Write. The role type applied to this end of the connector.                                                                                                                               |                                                        |
| <b>Stereotype</b>        | <i>String</i>                                       | Read/Write. Sets or gets the stereotype for this connector end.                                                                                                                               |                                                        |
| <b>StereotypeEx</b>      | <i>String</i>                                       | Read/Write. All the applied stereotypes of the connector end in a comma-separated list.                                                                                                       |                                                        |
| <b>TaggedValues</b>      | <i>Private</i>                                      | Read only. Collection of <b>RoleTag</b> objects.                                                                                                                                              | <a href="#">RoleTag Class</a><br><small>[1919]</small> |
| <b>Visibility</b>        | <i>String</i>                                       | Read/Write. Scope associated with this connector end. Valid types are: <i>Public</i> , <i>Private</i> , <i>Protected</i> and <i>Package</i> .                                                 |                                                        |

#### ConnectorEnd Methods

| Method                 | Type          | Notes                                                                                                                                                                         |
|------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error |

| Method           | Type           | Notes                                                                                                                                                                               |
|------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  |                | occurs.                                                                                                                                                                             |
| <b>Update ()</b> | <i>Boolean</i> | Update the current <i>ConnectorEnd</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### 18.2.2.7.4 ConnectorTag Class

A *ConnectorTag* is a Tagged Value for a connector and is accessed through the Connector *TaggedValues* collection.

Associated table in .EAP file: *t\_connectortag*

#### ConnectorTag Attributes

| Attribute          | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ConnectorID</b> | <i>Long</i>                          | Read/Write. The local ID of the associated connector.                                                                                                                                                                                                                                                                                                                                                                        |
| <b>FQName</b>      | <i>String</i>                        | Read only. The fully qualified name of the tag.                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Name</b>        | <i>String</i>                        | Read/Write. The tag or name.                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Notes</b>       | <i>String</i>                        | Read/Write. Further descriptive notes about this tag.<br>If <b>Value</b> (below) is set to " <b>&lt;memo&gt;</b> ", then <b>Notes</b> should contain the actual Tagged Value content.                                                                                                                                                                                                                                        |
| <b>ObjectType</b>  | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                                                                                                                                                                                                    |
| <b>TagGUID</b>     | <i>String</i>                        | Read/Write. A globally unique ID for this Tagged Value.                                                                                                                                                                                                                                                                                                                                                                      |
| <b>TagID</b>       | <i>Long</i>                          | Read only. A local ID to identify the Tagged Value.                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Value</b>       | <i>String</i>                        | Read/Write. The value assigned to this tag.<br>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to " <b>&lt;memo&gt;</b> " and insert the body of text in the <b>Notes</b> attribute (above).<br>When reading existing Tagged Values, if <b>Value</b> = " <b>&lt;memo&gt;</b> " then the developer should read the actual body of text from the <b>Notes</b> attribute. |

#### ConnectorTag Methods

| Method                 | Type           | Notes                                                                                                                                                                             |
|------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the current <i>ConnectorTag</i> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i>                              |

| Method | Type | Notes                          |
|--------|------|--------------------------------|
|        |      | function for more information. |

### 18.2.2.7.5 RoleTag Class

#### Reference:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b> | <p>This interface provides access to the association Role Tagged Values. Each connector end has a RoleTag collection that can be accessed to add, delete and access the RoleTags.</p> <p>In code you create something that resembles the following (where con is a Connector Object):</p> <p>Code fragment for accessing a RoleTag in VB.NET:</p> <pre> client = con.ClientEnd client.Role = "m_client" client.Update() tag = client.TaggedValues.AddNew ("tag", "value") tag.Update() tag = client.TaggedValues.AddNew ("tag2", "value2") tag.Update() client.TaggedValues.Refresh() For idx = 0 To client.TaggedValues. Count - 1     tag = client.TaggedValues.Get At (idx)     Console.WriteLine(tag.Tag) client.TaggedValues.DeleteAt (idx, False) Next tag = Nothing </pre> |          |

#### RoleTag Attributes

| Attribute           | Type                                                | Notes                                                                                               |
|---------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <b>BaseClass</b>    | <i>String</i>                                       | Read/Write. Indicates the role end; set to <b>ASSOCIATION_SOURCE</b> or <b>ASSOCIATION_TARGET</b> . |
| <b>ElementGUID</b>  | <i>String</i>                                       | Read/Write. GUID of the connector with which this role tag is associated.                           |
| <b>FQName</b>       | <i>String</i>                                       | Read only. The fully qualified name of the tag.                                                     |
| <b>ObjectType</b>   | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                           |
| <b>PropertyGUID</b> | <i>String</i>                                       | Read/Write. A system generated GUID to identify the Tagged Value.                                   |
| <b>Tag</b>          | <i>String</i>                                       | Read/Write. The actual tag name.                                                                    |
| <b>Value</b>        | <i>String</i>                                       | Read/Write. The value associated with this tag.                                                     |

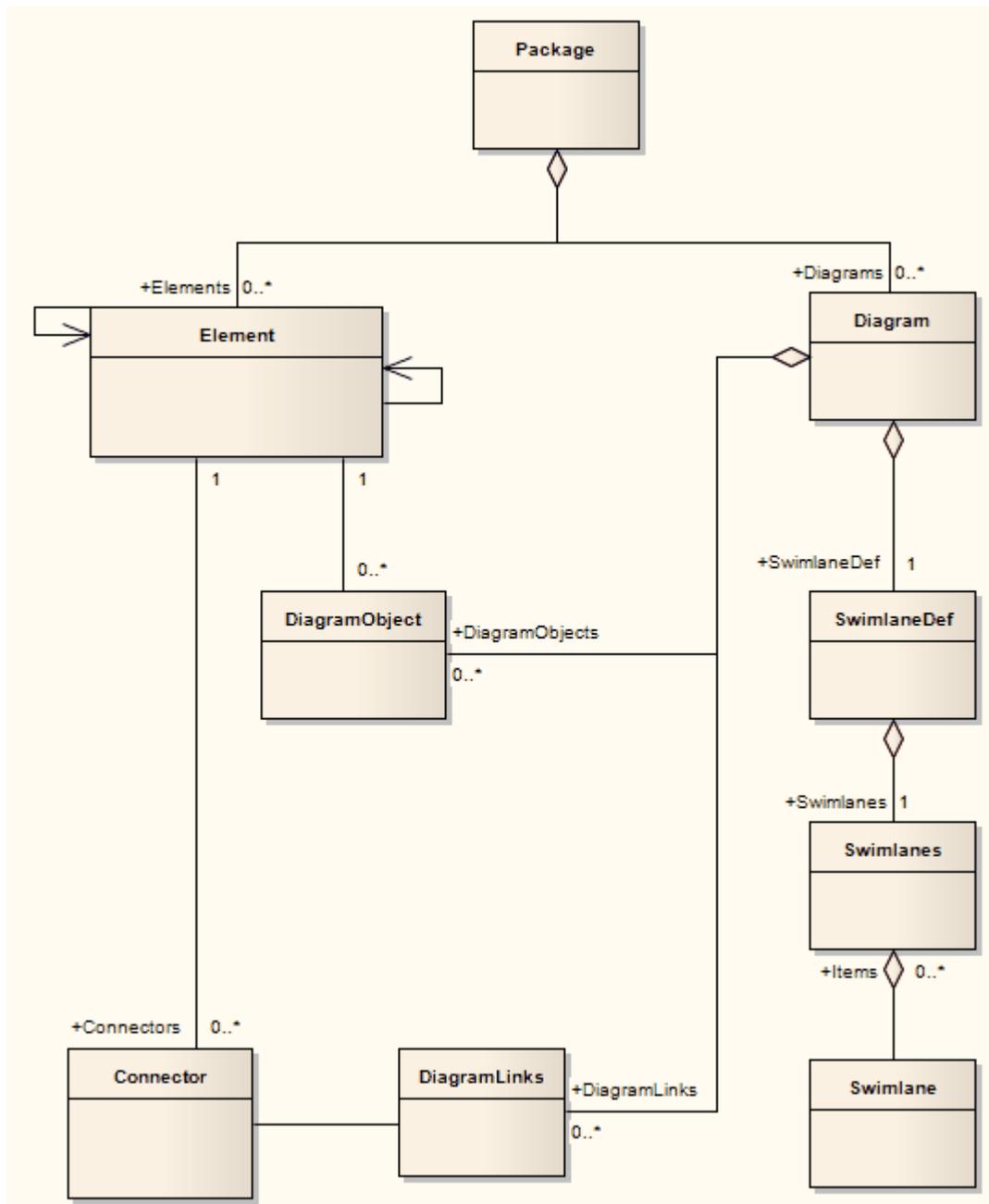
**RoleTag Methods**

| Method                 | Type           | Notes                                                                                                                                                                                 |
|------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ()</b> | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs. |
| <b>Update ()</b>       | <i>Boolean</i> | Update the RoleTag after changes or on initial creation. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.                                    |

**18.2.2.8 Diagram Package**

The *Diagram* package has information on a diagram and on *DiagramObject* and *DiagramLink*, which are the instances of elements within a diagram.

**Example:**



### 18.2.2.8.1 Diagram Class

#### Topics:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                     | See also |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>General Usage</b> | A <i>Diagram</i> corresponds to a single Enterprise Architect diagram. It is accessed through the Package Diagrams collection and in turn contains a collection of diagram objects and diagram connectors. Adding to the <i>Diagram Object</i> Class adds an element to the diagram (the element must already exist). When adding a new diagram, you must set the diagram type to a valid type; these are: |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                          | See also |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <ul style="list-style-type: none"> <li>• Activity</li> <li>• Analysis</li> <li>• Component</li> <li>• Custom</li> <li>• Deployment</li> <li>• Logical</li> <li>• Sequence</li> <li>• Statechart</li> <li>• Use Case</li> </ul> <p>Use the Analysis type for a Collaboration Diagram.</p> <p>Associated table in .EAP file: <i>t_diagram</i></p> |          |

### Diagram Attributes

| Attribute               | Type                                                            | Notes                                                                                                                                                                                                                                                                                                                                                         | See Also                                                     |
|-------------------------|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| <b>Author</b>           | <i>String</i>                                                   | Read/Write. The author.                                                                                                                                                                                                                                                                                                                                       |                                                              |
| <b>CreatedDate</b>      | <i>Date</i>                                                     | Read/Write. The date the diagram was created.                                                                                                                                                                                                                                                                                                                 |                                                              |
| <b>cx</b>               | <i>Long</i>                                                     | Read/Write. The X dimension of the diagram (default is 800).                                                                                                                                                                                                                                                                                                  |                                                              |
| <b>cy</b>               | <i>Long</i>                                                     | Read/Write. The Y dimension of the diagram (default is 1100).                                                                                                                                                                                                                                                                                                 |                                                              |
| <b>DiagramGUID</b>      | <i>Variant</i>                                                  | Read/Write. A globally unique ID for this diagram.                                                                                                                                                                                                                                                                                                            |                                                              |
| <b>DiagramID</b>        | <i>Long</i>                                                     | Read only. A local ID for the diagram.                                                                                                                                                                                                                                                                                                                        |                                                              |
| <b>DiagramLinks</b>     | <a href="#">Collection</a><br><small>n<sup>[1847]</sup></small> | <p>Read only. A list of <b>DiagramLink</b> objects, each containing information about the display characteristics of a connector in a diagram.</p> <p>A <b>DiagramLink</b> is only created once a user modifies a connector in a diagram in some way. Until this condition has been met default values are used and the <b>DiagramLink</b> is not in use.</p> | <a href="#">DiagramLink Class</a><br><small>[1924]</small>   |
| <b>DiagramObjects</b>   | <a href="#">Collection</a><br><small>n<sup>[1847]</sup></small> | Read only. A collection of references to <b>DiagramObjects</b> . A <b>DiagramObject</b> is an instance of an element in a diagram, and includes size and display characteristics.                                                                                                                                                                             | <a href="#">DiagramObject Class</a><br><small>[1925]</small> |
| <b>ExtendedStyle</b>    | <i>String</i>                                                   | Read/Write. An extended style attribute.                                                                                                                                                                                                                                                                                                                      |                                                              |
| <b>HighlightImports</b> | <i>Boolean</i>                                                  | Read/Write. Flag to indicate elements from other packages should be highlighted.                                                                                                                                                                                                                                                                              |                                                              |
| <b>IsLocked</b>         | <i>Boolean</i>                                                  | Read/Write. Flag indicating whether this diagram is locked or not.                                                                                                                                                                                                                                                                                            |                                                              |
| <b>MetaType</b>         | <i>String</i>                                                   | Read only. The diagram's domain-specific meta type, as defined by an MDG Technology.                                                                                                                                                                                                                                                                          |                                                              |
| <b>ModifiedDate</b>     | <i>Variant</i>                                                  | Read/Write. The date the diagram was last modified.                                                                                                                                                                                                                                                                                                           |                                                              |

| Attribute           | Type                                               | Notes                                                                                                                                                                                                                                     | See Also                                                |
|---------------------|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| Name                | String                                             | Read/Write. The diagram name.                                                                                                                                                                                                             |                                                         |
| Notes               | String                                             | Read/Write. Set/retrieve notes for this diagram.                                                                                                                                                                                          |                                                         |
| ObjectType          | <a href="#">ObjectTy<br/>pe</a> <sup>[1835]</sup>  | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                                                                                                                 |                                                         |
| Orientation         | String                                             | Read/Write. Page orientation: <b>P</b> for Portrait or <b>L</b> for Landscape.                                                                                                                                                            |                                                         |
| PackageID           | Long                                               | Read/Write. An ID of the package that this diagram belongs to.                                                                                                                                                                            |                                                         |
| ParentID            | Long                                               | Read/Write. An optional ID of an element that 'owns' this diagram; for example, a Sequence diagram owned by a Use Case.                                                                                                                   |                                                         |
| Scale               | Long                                               | Read/Write. The zoom scale (default is 100).                                                                                                                                                                                              |                                                         |
| SelectedConnector   | <a href="#">Connect<br/>or</a> <sup>[1911]</sup>   | Read/Write. The currently selected connector on this diagram. Null if there is no currently selected diagram.                                                                                                                             |                                                         |
| SelectedObjects     | <a href="#">Collectio<br/>n</a> <sup>[1841]</sup>  | Read only. Gets a collection representing the currently selected elements on the diagram. Can remove objects from this collection to deselect them, and add elements to the collection by passing the Object ID as a name to select them. |                                                         |
| ShowDetails         | Long                                               | Read/Write. Flag to indicate Diagram Details text should be shown. <b>1</b> = Show, <b>0</b> = Hide.                                                                                                                                      |                                                         |
| ShowPackageContents | Boolean                                            | Read/Write. Flag to indicate package contents should be shown in the current diagram.                                                                                                                                                     |                                                         |
| ShowPrivate         | Boolean                                            | Read/Write. Flag to show or hide Private features.                                                                                                                                                                                        |                                                         |
| ShowProtected       | Boolean                                            | Read/Write. Flag to show or hide Protected features.                                                                                                                                                                                      |                                                         |
| ShowPublic          | Boolean                                            | Read/Write. Flag to show or hide Public features.                                                                                                                                                                                         |                                                         |
| Stereotype          | String                                             | Read/Write. Sets or gets the stereotype for this diagram.                                                                                                                                                                                 |                                                         |
| StyleEx             | String                                             | Read/Write. Advanced style settings. Reserved for the use of Sparx Systems.                                                                                                                                                               |                                                         |
| Swimlanes           | String                                             | Read/Write. Information on swimlanes contained in the diagram. Please note that this property is superseded by <b>SwimlaneDef</b> .                                                                                                       | <a href="#">SwimlaneDef<br/>Class</a> <sup>[1927]</sup> |
| SwimlaneDef         | <a href="#">Swimlan<br/>eDef</a> <sup>[1927]</sup> | Read/Write. Information on swimlanes contained in the diagram.                                                                                                                                                                            |                                                         |
| Type                | String                                             | Read only. The diagram type. See the <i>t_diagramtypes</i> table in the .EAP file for more information.                                                                                                                                   |                                                         |
| Version             | String                                             | Read/Write. The version of the diagram.                                                                                                                                                                                                   |                                                         |

### Diagram Methods

| Method                                                    | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See Also                            |
|-----------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| <b>ApplyGroupLock (string aGroupName)</b>                 | <i>Boolean</i> | Applies a group lock to this diagram object, for the specified group, on behalf of the current user.<br><br>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.<br><br>Parameter: <ul style="list-style-type: none"> <li>aGroupName: String - the name of the user group for which to set the group lock</li> </ul>                                                                                                                                                                                                                                            |                                     |
| <b>ApplyUserLock ()</b>                                   | <i>Boolean</i> | Applies a user lock to this diagram object, for the current user.<br><br>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                     |
| <b>GetLastError ()</b>                                    | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.                                                                                                                                                                                                                                                                                                                                                                                                                           |                                     |
| <b>ReleaseUserLock ()</b>                                 | <i>Boolean</i> | Releases a group lock or user lock on this diagram object.<br><br>Throws an exception if the operation fails. Use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                     |
| <b>ReorderMessages ()</b>                                 | <i>Void</i>    | Resets the display order of Sequence and Collaboration messages. Typically used after inserting or deleting messages in the diagram.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                     |
| <b>ShowAsElement List (bool ShowAsList, bool Persist)</b> | <i>Boolean</i> | Toggles the diagram display between diagram format and <b>Diagram List</b> depending on the value of <i>ShowAsList</i> .<br><br>If <i>Persist</i> is set, the display format is written to the database so the diagram always opens in that format (diagram or list). Otherwise, the display format falls back to the default (diagram) once the display is closed.<br><br>Parameters: <ul style="list-style-type: none"> <li>ShowAsList: Boolean - indicates diagram or <b>Diagram List</b></li> <li>Persist: Boolean - indicates set (maintain <i>ShowAsList</i> value) or not (revert to default)</li> </ul> | <a href="#">Diagram List</a><br>462 |
| <b>Update ()</b>                                          | <i>Boolean</i> | Updates this diagram object after modification or appending a new item.<br><br>If <b>false</b> is returned, use <i>GetLastError()</i> to retrieve error information.                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                     |

### 18.2.2.8.2 DiagramLinks Class

A *DiagramLink* is an object that holds display information about a connector between two elements in a specific diagram. It includes, for example, the custom points and display appearance. It can be accessed from the Diagram **DiagramLinks** collection.

Associated table in .EAP file: *t\_diagramlinks*

#### DiagramLinks Attributes

| Attribute   | Type                                 | Notes                                                                           |
|-------------|--------------------------------------|---------------------------------------------------------------------------------|
| ConnectorID | Long                                 | Read/Write. The ID of the associated connector.                                 |
| DiagramID   | Long                                 | Read/Write. The local ID for the associated diagram.                            |
| Geometry    | String                               | Read/Write. The geometry associated with the current connector in this diagram. |
| InstanceID  | Long                                 | Read only. Holds the connector identifier for the current model.                |
| IsHidden    | Boolean                              | Read/Write. Flag to indicate if this item is hidden or not.                     |
| ObjectType  | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.       |
| Path        | String                               | Read/Write. The path of the connector in this diagram.                          |
| Style       | String                               | Read/Write. Additional style information; for example, color, thickness.        |

#### DiagramLinks Methods

| Method          | Type    | Notes                                                                                                                                                                                     |
|-----------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.     |
| Update ()       | Boolean | Update the current <i>DiagramLink</i> object after modification or appending a new item.<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information. |

#### Learn More:

- [DiagramLinks](#) [1922]

#### 18.2.2.8.3 *DiagramObject Class*

The *DiagramObject* Class stores presentation information that indicates what is displayed in a diagram and how it is shown.

Associated table in .EAP file: *t\_diagramobjects*

#### DiagramObject Attributes

| Attribute  | Type | Notes                                                                   | See Also |
|------------|------|-------------------------------------------------------------------------|----------|
| Bottom     | Long | Read/Write; the bottom position of the element                          |          |
| DiagramID  | Long | Read/Write; the ID of the associated diagram (long)                     |          |
| ElementID  | Long | Read/Write; the <i>ElementID</i> of the object instance in this diagram |          |
| InstanceID | Long | Read/Write; read only attribute that holds the connector                |          |

| Attribute  | Type                                                           | Notes                                                                                                                                                   | See Also                                                   |
|------------|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|            |                                                                | identifier for the current model                                                                                                                        |                                                            |
| Left       | Long                                                           | Read/Write; the left position of the element                                                                                                            |                                                            |
| ObjectType | <a href="#">ObjectTyp</a><br><small>e<sup>[1835]</sup></small> | Read only; distinguishes objects referenced through a Dispatch interface                                                                                |                                                            |
| Right      | Long                                                           | Read/Write; the right position of the element                                                                                                           |                                                            |
| Sequence   | Long                                                           | Read/Write; the sequence position when loading into diagram (affects Z order)<br><br>The Z-order is one-based and the lowest value is in the foreground |                                                            |
| Style      | Variants                                                       | Write only (reading this value gives undefined results); style information for this object                                                              | <a href="#">Setting the Style</a><br><small>[1926]</small> |
| Top        | Long                                                           | Read/Write; the top position of the element                                                                                                             |                                                            |

#### DiagramObject Methods

| Method          | Type    | Notes                                                                                                                                                                               |
|-----------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GetLastError () | String  | Returns a string value describing the most recent error that occurred in relation to this object<br><br>This function is rarely used as an exception is thrown when an error occurs |
| Update ()       | Boolean | Update the current <i>Diagram Object</i> after modification or appending a new item<br><br>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information |

#### Topics:

| Topic             | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Setting The Style | <p>The <i>Style</i> attribute is used for setting the appearance of a <i>DiagramObject</i>; it is set with a string value in the format:</p> <p>BCol =n; BFol =n; LCol =n; LWth =n;</p> <p>where:</p> <ul style="list-style-type: none"> <li>• <i>BCol</i> = Background Color</li> <li>• <i>BFol</i> = Font Color</li> <li>• <i>LCol</i> = Line Color</li> <li>• <i>LWth</i> = Line Width</li> </ul> <p>The color value is a decimal representation of the hex RGB value, where Red=FF, Green=FF00 and Blue=FF0000; for example:</p> <p>DiagramObject.Style = "BCol =35723; BFol =9342520; LCol =9342520; LWth =1; "</p> |          |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                  | See also |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>The following code snippet shows how you might change the style settings for all of the objects in the current diagram, in this case changing everything to red:</p> <pre> For Each aDiagramObj In aDiagram.DiagramObjects     aDiagramObj.Style = "BColor=255; BFol=9342520; LCol=9342520; LWh=1;"     aDiagramObj.Update aRepos.ReloadDiagram aDiagramObj.  DiagramL D Next </pre> |          |

#### 18.2.2.8.4 SwimlaneDef Class

A *SwimlaneDef* object makes available attributes relating to a single row or column in a list of swimlanes.

| Attribute             | Type                                 | Notes                                                                                                         |
|-----------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------|
| <b>Bold</b>           | <i>Boolean</i>                       | Read/Write. Show the title text in bold.                                                                      |
| <b>FontColor</b>      | <i>Long</i>                          | Read/Write. RGB color used to draw the titles.                                                                |
| <b>HideClassifier</b> | <i>Boolean</i>                       | Read/Write. Removes any classifier from title display.                                                        |
| <b>HideNames</b>      | <i>Boolean</i>                       | Read/Write. Set to true to hide the swimlane titles.                                                          |
| <b>LineColor</b>      | <i>Long</i>                          | Read/Write. RGB color used to draw swimlane borders.                                                          |
| <b>LineWidth</b>      | <i>Long</i>                          | Read/Write. Width of line, in pixels, used to draw swimlanes. Valid values: <b>1</b> , <b>2</b> or <b>3</b> . |
| <b>Locked</b>         | <i>Boolean</i>                       | Read/Write. If set to true, disables user modification of the swimlanes via the diagram.                      |
| <b>ObjectType</b>     | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface.                                     |
| <b>Orientation</b>    | <i>String</i>                        | Read/Write. Indication of whether the swimlanes are vertical or horizontal.                                   |
| <b>ShowInTitleBar</b> | <i>Boolean</i>                       | Read/Write. Enables vertical swimlane titles to be shown in title bar.                                        |
| <b>Swimlanes</b>      | <a href="#">Swimlanes</a><br>[1927]  | Read/Write. A list of individual swimlanes.                                                                   |

#### 18.2.2.8.5 Swimlanes Class

A *Swimlanes* object is attached to a diagram's **SwimlaneDef** object and provides a mechanism to access individual swimlanes.

##### Swimlanes Attributes

| Attribute         | Type                                 | Notes                                                                     |
|-------------------|--------------------------------------|---------------------------------------------------------------------------|
| <b>Count</b>      | <i>Long</i>                          | Read/Write. Gives the number of swimlanes.                                |
| <b>ObjectType</b> | <a href="#">ObjectType</a><br>[1835] | Read only. Distinguishes objects referenced through a Dispatch interface. |

Swimlanes Methods

| Method                                               | Type                                                                   | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Add (string Title, long Width)</b>                | <a href="#">Swimlane</a><br><small>[1928]</small>                      | Adds a new swimlane to the end of the list. Returns a swimlane object representing the newly added entry.<br><br>Parameters: <ul style="list-style-type: none"> <li>Title: String - The title text that appears at the top of the swimlane. Can be the same as an existing swimlane title.</li> <li>Width: Long - The width of the swimlane in pixels.</li> </ul>                                                                                                               |
| <b>Delete (object Index)</b>                         | <i>Void</i>                                                            | Deletes a selected swimlane.<br><br>If the string matches more than one entry, only the first entry is deleted.<br><br>Parameter: <ul style="list-style-type: none"> <li>Index: Object - Either a string representing the title text or an integer representing the zero-based index of the swimlane to delete.</li> </ul>                                                                                                                                                      |
| <b>DeleteAll ()</b>                                  | <i>Void</i>                                                            | Removes all swimlanes.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Insert (long Index, string Title, long Width)</b> | <a href="#">Swimlane</a><br><small>[1928]</small>                      | Inserts a swimlane at a specific position. Returns a swimlane object representing the newly added entry.<br><br>Parameters: <ul style="list-style-type: none"> <li>Index: Long - The zero-based index of the existing Swimlane before which this new entry is inserted.</li> <li>Title: String - The title text which appears at the top of the swimlane. Can be the same as an existing swimlane title.</li> <li>Width: Long - The width of the swimlane in pixels.</li> </ul> |
| <b>Items (object Index)</b>                          | <a href="#">Swimlane</a><br><small>[1928]</small><br><i>collection</i> | Accesses an individual swimlane.<br><br>If the string matches more than one swimlane title, the first matching swimlane is returned.<br><br>Parameter: <ul style="list-style-type: none"> <li>Index: Object - Either a string representing the title text or an integer representing the zero-based index of the swimlane to get.</li> </ul>                                                                                                                                    |

Learn More:

- [SwimlaneDef Class](#) [1927]

**18.2.2.8.6 Swimlane Class**

A *Swimlane* object makes available attributes relating to a single row or column in a list of **swimlanes**.

| Attribute             | Type                                                | Notes                                                                                                                                         |
|-----------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>BackColor</b>      | <i>Long</i>                                         | Read/Write. The swimlane is filled with this RGB color.                                                                                       |
| <b>ClassifiedGuid</b> | <i>String</i>                                       | Read/Write. The GUID of the classifier Class. This can be obtained from the corresponding Element object via the <i>ElementGUID</i> property. |
| <b>ObjectType</b>     | <a href="#">ObjectType</a><br><small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface.                                                                     |

| Attribute    | Type          | Notes                                            |
|--------------|---------------|--------------------------------------------------|
| <b>Title</b> | <i>String</i> | Read/Write. Text at the head of the swimlane.    |
| <b>Width</b> | <i>Long</i>   | Read/Write. The width of the swimlane in pixels. |

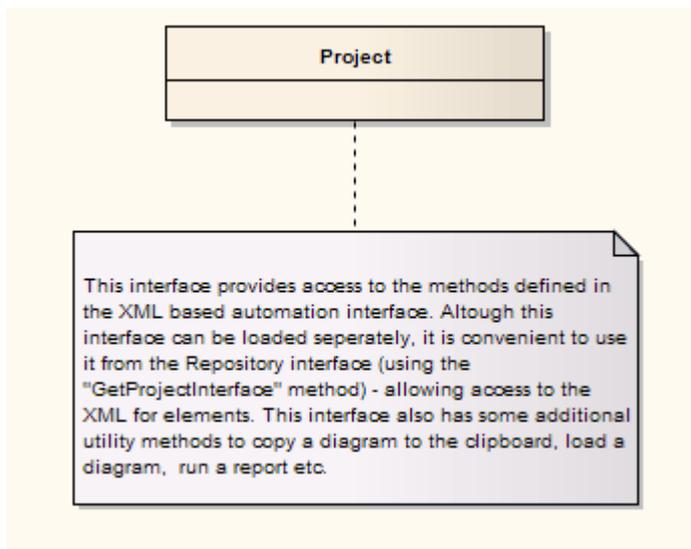
Learn More:

- [Swimlane Class](#) [1927]

### 18.2.2.9 Project Interface Package

The *Enterprise Architect.Project* interface. This is the XML-based interface to Enterprise Architect elements; it also includes some utility functions. You can get a pointer to this interface using the *Repository*. *GetProjectInterface* method.

Example:



#### 18.2.2.9.1 Project Class

The Project interface can be accessed from the Repository using *GetProjectInterface()*. The returned interface provides access to the XML-based Enterprise Architect Automation Interface. Use this interface to get XML for the various internal elements and to run some utility functions to perform tasks such as load diagrams or run reports.

Project Attributes

| Attribute         | Type                                             | Notes                                                                      |
|-------------------|--------------------------------------------------|----------------------------------------------------------------------------|
| <b>ObjectType</b> | <a href="#">ObjectType</a> <small>[1835]</small> | Read only<br>Distinguishes objects referenced through a Dispatch interface |

## Project Methods

| Method                                                                                                  | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See Also                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CreateBaseline</b> (string PackageGUID, string Version, string Notes)                                | <i>Boolean</i> | Creates a Baseline of a specified package<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to Baseline</li> <li>Version: String - the version of the Baseline</li> <li>Notes: String - any notes concerning the Baseline</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                     |
| <b>CreateBaselineEx</b> (string PackageGUID, string Version, string Notes, EA.CreateBaselineFlag Flags) | <i>Boolean</i> | Creates a Baseline of a specified package, with a flag to <b>exclude package contents</b> below the first level<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to Baseline</li> <li>Version: String - the version of the Baseline</li> <li>Notes: String - any notes concerning the Baseline</li> <li>Flags: <b>EA.CreateBaselineFlag</b> - whether or not to exclude package contents below the first level</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <a href="#">Create Baselines</a> <sup>[312]</sup><br><a href="#">EA.CreateBaselineFlag</a> <sup>[1832]</sup>                                                                                                                        |
| <b>DefineRule</b> (string CategoryID, EA.EnumMVErrType ErrorType, string ErrorMessage)                  | <i>String</i>  | Defines the individual rules that can be performed during model validation. It must be called once for each rule from the <b>EA_OnInitializeUserRules</b> broadcast handler.<br><br>The return value is a <i>RuleId</i> , which can be used for reference purposes when an individual rule is executed by Enterprise Architect during model validation.<br><br>See <b>Model Validation Example</b> for a detailed example of use of this method.<br>Parameters: <ul style="list-style-type: none"> <li>CategoryId: String - should be passed the return value from the <b>DefineRuleCategory</b> method.</li> <li>ErrorType: EA.EnumMVErrType - depending on the severity of the error being validated, can be: <ul style="list-style-type: none"> <li><b>mvErrorCritical</b></li> <li><b>mvError</b></li> <li><b>mvWarning</b>, or</li> <li><b>mvInformation</b>.</li> </ul> </li> <li>ErrorMessage: String - can contain a default error string, although this is probably overridden by the</li> </ul> | <a href="#">EA_OnInitializeUserRules</a> <sup>[2001]</sup><br><a href="#">Model Validation Example</a> <sup>[2006]</sup><br><a href="#">DefineRuleCategory</a> <sup>[1937]</sup><br><a href="#">PublishResult</a> <sup>[1943]</sup> |

| Method                                                                                                          | Type    | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See Also                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                 |         | <b>PublishResult</b> call.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                          |
| <b>DefineRuleCategory</b><br>(string CategoryName)                                                              | String  | <p>Defines a category of rules that can be performed during model validation (there is typically one category per Add-In). It must be called once <i>from the</i> <b>EA_OnInitializeUserRules</b> broadcast handler.</p> <p>The return value is a CategoryId that must to be passed to the <b>DefineRule</b> method.</p> <p>See <b>Model Validation Example</b> for a detailed example of use of this method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• CategoryName: String - a text string that is visible in the <b>Model Validation Configuration</b> dialog.</li> </ul>                                                                                                                                                                                                                                                                                                                  | <a href="#">EA_OnInitializeUserRules</a> <sup>[2000]</sup><br><a href="#">DefineRule</a> <sup>[1930]</sup><br><a href="#">Model Validation Example</a> <sup>[2006]</sup> |
| <b>DeleteBaseline</b> (string BaselineGUID)                                                                     | Boolean | <p>Deletes a Baseline, identified by the BaselineGUID, from the repository.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• BaselineGUID: String - the GUID (in XML format) of the Baseline to delete.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                          |
| <b>DoBaselineCompare</b><br>(string PackageGUID, string Baseline, string ConnectString)                         | String  | <p>Performs a Baseline comparison using the supplied package GUID and Baseline GUID (obtained in the result list <i>from</i> <b>GetBaselines</b>).</p> <p>Optionally you can include the connection string required to find the Baseline if it exists in a different model file.</p> <p>This method returns a log file of the status of all elements found and compared in the difference procedure. You can use this log information as input to <b>DoBaselineMerge</b> - automatically merging information from the Baseline.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package to run the comparison on.</li> <li>• Baseline: String - the GUID (in XML format) of the Baseline to run the comparison on.</li> <li>• ConnectString: String - the location of the external .EAP file or DBMS to extract the Baseline from.</li> </ul> | <a href="#">GetBaselines</a> <sup>[1937]</sup><br><a href="#">DoBaselineMerge</a> <sup>[1931]</sup>                                                                      |
| <b>DoBaselineMerge</b><br>(string PackageGUID, string Baseline, string MergeInstructions, string ConnectString) | String  | <p>Performs a batch merge based on instructions contained in an XML file (<i>MergeInstructions</i>). You can supply an optional connection string if the Baseline is located in another model.</p> <p>In the <i>MergeInstructions</i> file, each</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <a href="#">DoBaselineCompare()</a> <sup>[1931]</sup>                                                                                                                    |

| Method                                          | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See Also |
|-------------------------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                 |                                      | <p><i>MergeItem</i> node supplies the GUID of a differenced item from the XML difference log. As the merge is uni-directional and actioned in only one possible way, no additional arguments are required. Enterprise Architect chooses the correct procedure based on the Difference results.</p> <pre>&lt;Merge&gt;     &lt;MergeItem guid="{XXXXXXXX}" /&gt;     &lt;MergeItem guid="{XXXXXXXX}" /&gt; &lt;/Merge&gt;</pre> <p>Alternatively, you can supply a single <i>MergeItem</i> with a GUID of <i>RestoreAll</i>. In this case, Enterprise Architect batch-processes ALL differences.</p> <pre>&lt;Merge&gt;     &lt;MergeItem guid="RestoreAll" changed="true" baselineOnly="true" modelOnly="true" moved="true" fullRestore="false" / &gt; &lt;/Merge&gt;</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to merge the Baseline into.</li> <li>Baseline: String - the GUID of the Baseline (in XML format) to merge into the package.</li> <li>MergeInstructions: String - the file containing the GUID of each differenced item from the XML difference log returned by <b>DoBaselineCompare()</b>.</li> <li>ConnectionString: String - the location of the EAP file or DBMS to get the Baseline from, if not in the same model as the package.</li> </ul> |          |
| <b>EnumDiagramElements (string DiagramGUID)</b> | protected abstract:<br><i>String</i> | <p>Gets an XML list of all elements in a diagram.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to get elements for.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |          |
| <b>EnumDiagrams (string PackageGUID)</b>        | protected abstract:<br><i>String</i> | <p>Gets an XML list of all diagrams in a specified package.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |

| Method                                                                                                                                                | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                     | See Also                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                       |                                      | XML format) of the package to list diagrams for.                                                                                                                                                                                                                                                                                                                          |                                                                                                                     |
| <b>EnumElements (string PackageGUID)</b>                                                                                                              | protected abstract:<br><i>String</i> | Gets an XML list of elements in a specified package.<br>Parameters:<br><ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to get a list of elements for.</li> </ul>                                                                                                                                                     |                                                                                                                     |
| <b>EnumLinks (string ElementGUID)</b>                                                                                                                 | protected abstract:<br><i>String</i> | Gets an XML list of connectors for a specified element.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to get all associated connectors for.</li> </ul>                                                                                                                                           |                                                                                                                     |
| <b>EnumPackages (string PackageGUID)</b>                                                                                                              | protected abstract:<br><i>String</i> | Gets an XML list of child packages inside a parent package.<br>Parameters:<br><ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the parent package.</li> </ul>                                                                                                                                                                     |                                                                                                                     |
| <b>EnumProjects ()</b>                                                                                                                                | protected abstract:<br><i>String</i> | Gets a list of projects in the current file; corresponds to <b>Models</b> in <i>Repository</i> .                                                                                                                                                                                                                                                                          | <a href="#">Models</a>         |
| <b>EnumViews ()</b>                                                                                                                                   | protected abstract:<br><i>String</i> | Enumerates the Views for a project. Returned as an XML document.                                                                                                                                                                                                                                                                                                          |                                                                                                                     |
| <b>EnumViewEx (string ProjectGUID)</b>                                                                                                                | protected abstract:<br><i>String</i> | Gets a list of Views in the current project.<br>Parameters:<br><ul style="list-style-type: none"> <li>ProjectGUID: String - the GUID (in XML format) of the project to get views for.</li> </ul>                                                                                                                                                                          |                                                                                                                     |
| <b>Exit ()</b>                                                                                                                                        | protected abstract:<br><i>String</i> | Exits the current instance of Enterprise Architect; this function is maintained for backward compatibility and should never be called.<br><br>Enterprise Architect automatically exits when you are no longer using any of the provided objects.                                                                                                                          |                                                                                                                     |
| <b>ExportPackageXML (string PackageGUID, enumXMIMType XMIMType, long DiagramXML, long DiagramImage, long FormatXML, long UseDTD, string FileName)</b> | protected abstract:<br><i>String</i> | Exports XMI for a specified package.<br>Parameters:<br><ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to be exported.</li> <li>XMIMType: EnumXMIMType - specifies the XMI type and version information; see <b>XMIMType Enum</b> for accepted values.</li> <li>DiagramXML: Long - <b>true</b> if XML for</li> </ul> | <a href="#">XMIMType Enum</a>  |

| Method                                                                                                                                                                                         | Type                                  | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also                                                                                                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                                |                                       | <p>diagrams is required; accepted values:</p> <ul style="list-style-type: none"> <li><b>0</b> = Do not export diagrams</li> <li><b>1</b> = Export diagrams</li> <li><b>2</b> = Export diagrams along with alternate images.</li> </ul> <ul style="list-style-type: none"> <li>• DiagramImage: Long - the format for diagram images to be created at the same time; accepted values: <ul style="list-style-type: none"> <li><b>-1</b>=NONE</li> <li><b>0</b>=EMF</li> <li><b>1</b>=BMP</li> <li><b>2</b>=GIF</li> <li><b>3</b>=PNG</li> <li><b>4</b>=JPG.</li> </ul> </li> <li>• FormatXML: Long - <b>true</b> if XML output should be formatted prior to saving.</li> <li>• UseDTD: Long - <b>true</b> if a DTD should be used.</li> <li>• FileName: String - the filename to output to.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                              |
| <p><b>ExportPackageXMLEx</b> (string PackageGUID, enumXMIMType XMIMType, long DiagramXML, long DiagramImage, long FormatXML, long UseDTD, string FileName, ea. ExportPackageXMLFlag Flags)</p> | <p>protected abstract:<br/>String</p> | <p>Exports XML for a specified package, with a flag to determine whether the export <b>includes package content</b> below the first level.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package to be exported.</li> <li>• XMIMType: EnumXMIMType - specifies the XML type and version information; see <b>XMIMType Enum</b> for accepted values.</li> <li>• DiagramXML: Long - <b>true</b> if XML for diagrams is required; accepted values: <ul style="list-style-type: none"> <li><b>0</b> = Do not export diagrams</li> <li><b>1</b> = Export diagrams</li> <li><b>2</b> = Export diagrams along with alternate images.</li> </ul> </li> <li>• DiagramImage: Long - the format for diagram images to be created at the same time; accepted values: <ul style="list-style-type: none"> <li><b>-1</b>=NONE</li> <li><b>0</b>=EMF</li> <li><b>1</b>=BMP</li> <li><b>2</b>=GIF</li> <li><b>3</b>=PNG</li> <li><b>4</b>=JPG.</li> </ul> </li> <li>• FormatXML: Long - <b>true</b> if XML output should be formatted prior to saving.</li> <li>• UseDTD: Long - <b>true</b> if a DTD</li> </ul> | <p><a href="#">Package Content</a> <sup>[329]</sup></p> <p><a href="#">XMIMType Enum</a> <sup>[1838]</sup></p> <p><a href="#">ExportPackageXMLFlag</a> <sup>[1834]</sup></p> |

| Method                                                                                                                      | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also                                                 |
|-----------------------------------------------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
|                                                                                                                             |                | <p>should be used.</p> <ul style="list-style-type: none"> <li>• <b>FileName</b>: String - the filename to output to.</li> <li>• <b>Flags</b>: <b>ea.ExportPackageXMIFlag</b> - whether or not to include package content below the first level (currently only supported for <i>xmiEADefault</i>).</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                          |
| <b>GenerateClass</b> (string ElementGUID, string ExtraOptions)                                                              | <i>Boolean</i> | <p>Generates the code for a single Class.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <b>ElementGUID</b>: String - the GUID (in XML format) of the element to generate.</li> <li>• <b>ExtraOptions</b>: String - enables extra options to be given to the command; currently unused.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                          |
| <b>GenerateDiagramFromScenario</b> (string ElementGUID, EnumScenarioDiagramType DiagramType, long OverwriteExistingDiagram) | <i>Boolean</i> | <p>Generates various diagrams from the Structured Specification of an element.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <b>ElementGUID</b>: String - the GUID (in XML format) of the element containing the Structured Specification.</li> <li>• <b>DiagramType</b>: EnumScenarioDiagramType - the type of diagram to generate; see <b>ScenarioDiagramType Enum</b> for accepted values</li> <li>• <b>OverwriteExistingDiagram</b>: Long - determines whether to overwrite the existing diagram or synchronize the existing elements with the scenario steps <ul style="list-style-type: none"> <li><b>0</b> = Delete existing diagram and elements, and create new diagram and elements</li> <li><b>1</b> = Synchronize existing elements with scenario steps and preserve diagram layout</li> <li><b>2</b> = Synchronize existing elements with scenario steps and re-cast diagram layout</li> <li><b>3</b> = Do not generate diagram if one already exists.</li> </ul> </li> </ul> | <a href="#">ScenarioDiagramType Enum</a> <sup>1837</sup> |
| <b>GenerateElementDDL</b> (string ElementGUID, string FileName, string ExtraOptions)                                        | <i>Boolean</i> | <p>Generates DDL for an element</p> <p>For example:</p> <pre>owner = true; pkfkconstraints = true; comment level = 3;</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <b>ElementGUID</b>: String – the GUID (in XML</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                          |

| Method                                                                               | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See Also |
|--------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                                      |                | <p>format) of the element to generate DDL for</p> <ul style="list-style-type: none"> <li>• FileName: String – the target file path to which to write the generated DDL</li> <li>• ExtraOptions: String – enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>• Generate Owner (<i>owner</i>)</li> <li>• Generate PK/FK Key Constraints (<i>pkfkconstraints</i>)</li> <li>• Generate Comments (<b>0</b> – None, <b>1</b> - Table, <b>2</b> - Column, <b>3</b> - All)</li> </ul> </li> </ul>                                                                               |          |
| <b>GeneratePackage</b> (string PackageGUID, string ExtraOptions)                     | <i>Boolean</i> | <p>Generates the code for all Classes within a package.</p> <p>For example:</p> <pre>recurse=1; overwrite=1; dir=C:\</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package to generate code for</li> <li>• ExtraOptions: String - enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>• Generation of all subpackages (<i>recurse</i>)</li> <li>• Force overwrite of all files (<i>overwrite</i>) and</li> <li>• Specification to auto generate all paths (<i>dir</i>).</li> </ul> </li> </ul> |          |
| <b>GeneratePackageDDL</b> (string PackageGUID, string FileName, string ExtraOptions) | <i>Boolean</i> | <p>Generates DDL for a package.</p> <p>For example:</p> <pre>owner=true; pkfkconstraints=true; commentlevel=3; singlefile=true; includechild=true;</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String – the GUID (in XML format) of the package to generate DDL for</li> <li>• FileName: String – the target file path to which to write the generated DDL</li> <li>• ExtraOptions: String – enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>• Generate DDL for child packages (</li> </ul> </li> </ul>                           |          |

| Method                                                                                    | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See Also                                                                                                                   |
|-------------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
|                                                                                           |                | <p><i>includechild</i>)</p> <ul style="list-style-type: none"> <li>• Generate Owner (<i>owner</i>)</li> <li>• Generate PK/FK Key Constraints (<i>pkfkconstraints</i>)</li> <li>• Generate Comments (<b>0</b> - None, <b>1</b> - Table, <b>2</b> - Column, <b>3</b> - All)</li> <li>• Generate in SingleFile (<i>singlefile</i>)</li> </ul>                                                                                                                                                                                                                                                                                                                                     |                                                                                                                            |
| <b>GenerateTestFromScenario</b> (string ElementGUID, EnumScenarioTestType TestType)       | <i>Boolean</i> | <p>Generates either an <b>Internal test</b> or an <b>External test</b> from the Structured Specification of an element.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ElementGUID: String - the GUID (in XML format) of the element containing the Structured Specification</li> <li>• TestType: EnumScenarioTestType - the type of test to generate; see <b>ScenarioTestType Enum</b> for accepted values</li> </ul>                                                                                                                                                                                                                                        | <p><a href="#">Generate Test Cases</a> <sup>[685]</sup></p> <p><a href="#">ScenarioTestType Enum</a> <sup>[1838]</sup></p> |
| <b>GenerateXSD</b> (string PackageGUID, string FileName, string Encoding, string Options) | <i>Boolean</i> | <p>Creates an XML schema for this GenerateXSD. Returns <b>true</b> on success.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package.</li> <li>• FileName: String - target filepath.</li> <li>• Encoding: String - the XML encoding for the code page instruction.</li> <li>• Options: String - enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>• <i>GenGlobalElement</i> - turn the generation of global elements for all global <i>ComplexTypes</i> <b>On</b> or <b>Off</b>; for example: - GenGlobalElement = 1.</li> </ul> </li> </ul> |                                                                                                                            |
| <b>GetBaselines</b> (string PackageGUID, string ConnectString)                            | <i>String</i>  | <p>Returns a list (in XML format) of Baselines associated with the supplied package GUID. Optionally, you can provide a connection string to get Baselines from the same package, but located in a different model file (or DBMS).</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package to get Baselines for.</li> <li>• ConnectString: String - the location of the EAP file or DBMS to get the Baselines from, if not in the same model as the package.</li> </ul>                                                                                                                                  |                                                                                                                            |

| Method                                             | Type                                 | Notes                                                                                                                                                                                                          | See Also |
|----------------------------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>GetDiagram (string DiagramGUID)</b>             | protected abstract:<br><i>String</i> | Gets diagram details, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to get details for.</li> </ul>                     |          |
| <b>GetElement (string ElementGUID)</b>             | protected abstract:<br><i>String</i> | Gets XML for the specified element.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to retrieve XML for.</li> </ul>                     |          |
| <b>GetElementConstraints (string ElementGUID)</b>  | protected abstract:<br><i>String</i> | Gets constraints for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                             |          |
| <b>GetElementEffort (string ElementGUID)</b>       | protected abstract:<br><i>String</i> | Gets efforts for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                                 |          |
| <b>GetElementFiles (string ElementGUID)</b>        | protected abstract:<br><i>String</i> | Gets metrics for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                                 |          |
| <b>GetElementMetrics (string ElementGUID)</b>      | protected abstract:<br><i>String</i> | Gets files for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                                   |          |
| <b>GetElementProblems (string ElementGUID)</b>     | protected abstract:<br><i>String</i> | Gets a list of issues (problems) associated with an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul> |          |
| <b>GetElementProperties (string ElementGUID)</b>   | protected abstract:<br><i>String</i> | Gets Tagged values for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                           |          |
| <b>GetElementRequirements (string ElementGUID)</b> | protected abstract:<br><i>String</i> | Gets a list of requirements for an element, in XML format.<br>Parameters:<br><ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                  |          |
| <b>GetElementResources (string ElementGUID)</b>    | protected abstract:                  | Gets a list of resources for an element, in XML format.                                                                                                                                                        |          |

| Method                                                                                                  | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also |
|---------------------------------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                                                         | <i>String</i>                        | Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                                                                                                                                                                                                                                                                                                            |          |
| <b>GetElementRisks (string ElementGUID)</b>                                                             | protected abstract:<br><i>String</i> | Gets a list of risks associated with an element, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                                                                                                                                                                                                                                         |          |
| <b>GetElementScenarios (string ElementGUID)</b>                                                         | protected abstract:<br><i>String</i> | Gets a list of scenarios for an element, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                                                                                                                                                                                                                                                 |          |
| <b>GetElementTests (string ElementGUID)</b>                                                             | protected abstract:<br><i>String</i> | Gets a list of tests for an element, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element.</li> </ul>                                                                                                                                                                                                                                                     |          |
| <b>GetLastError ()</b>                                                                                  | protected abstract:<br><i>String</i> | Returns a string value describing the most recent error that occurred in relation to this object.<br><br>This function is rarely used as an exception is thrown when an error occurs.                                                                                                                                                                                                                                                   |          |
| <b>GetLink (string LinkGUID)</b>                                                                        | protected abstract:<br><i>String</i> | Gets connector details, in XML format.<br>Parameters: <ul style="list-style-type: none"> <li>LinkGUID: String - the GUID (in XML format) of the connector to get details of.</li> </ul>                                                                                                                                                                                                                                                 |          |
| <b>GUIDtoXML (string GUID)</b>                                                                          | <i>String</i>                        | Changes an internal GUID to the form used in XML.<br>Parameters: <ul style="list-style-type: none"> <li>GUID: String - the Enterprise Architect style GUID to convert to XML format.</li> </ul>                                                                                                                                                                                                                                         |          |
| <b>ImportDirectory (string PackageGUID, string Language, string DirectoryPath, string ExtraOptions)</b> | <i>Boolean</i>                       | Imports a source code directory into the model.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to reverse engineer code into.</li> <li>Language: String - specifies the language of the code to be imported.</li> <li>DirectoryPath: String - specifies the path where the code is found on the computer.</li> <li>ExtraOptions: String - enables extra</li> </ul> |          |

| Method                                                                                             | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also |
|----------------------------------------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                                                    |                | options to be given to the command; currently enables import of source from all child directories ( <i>recurse</i> ) - for example: <i>recurse=1</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |          |
| <b>ImportFile</b> (string PackageGUID, string Language, string FileName, string ExtraOptions)      | <i>Boolean</i> | Imports an individual file or binary module into the model, in a package per namespace style import.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to reverse engineer code into; this is expected to be a namespace root package.</li> <li>Language: String - specifies the language of the code to be imported.</li> </ul> Use the value " <b>DNPE</b> " to import a binary module. This imports a .Net assembly or Java .class file, but <b>not</b> a .jar file. <ul style="list-style-type: none"> <li>Filename: String - specifies the path where the code or module is found on the computer.</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently unused.</li> </ul> |          |
| <b>ImportPackageXML</b> (string PackageGUID, string Filename, long ImportDiagrams, long StripGUID) | <i>String</i>  | Imports an XML file at a point in the tree.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the target package to import the XML file into (or overwrite with the XML file).</li> <li>Filename or XMLText: String - the name of the XML file</li> </ul> If the String is of type <i>filename</i> it is interpreted as a source file, otherwise the String is imported as XML text. <ul style="list-style-type: none"> <li>ImportDiagrams: Long.</li> <li>StripGUID: Long - boolean value to indicate whether to replace the element UniqueIDs on import; if stripped, then a package could be imported twice into Enterprise Architect, as two different versions.</li> </ul>                                                    |          |
| <b>LayoutDiagram</b> (string DiagramGUID, long LayoutStyle)                                        | <i>Boolean</i> | <b>Deprecated.</b> it is recommended that <i>LayoutDiagramEx</i> is used instead.<br>Calls the function to automatically layout a diagram in hierarchical fashion. It is only recommended for Class and Object diagrams.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |          |

| Method                                                                                                                                       | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
|                                                                                                                                              |                | Parameters: <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to lay out.</li> <li>LayoutStyle: Long - always ignored.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                          |
| <b>LayoutDiagramEx (string DiagramGUID, long LayoutStyle, long Iterations, long LayerSpacing, long ColumnSpacing, boolean SaveToDiagram)</b> | <i>Boolean</i> | Calls the function to automatically layout a diagram in hierarchical fashion. It is only recommended for Class and Object diagrams.<br><br><i>LayoutStyle</i> accepts the following options <ul style="list-style-type: none"> <li>Default Options:               <ul style="list-style-type: none"> <li>IsDiagramDefault</li> <li>IsProgramDefault.</li> </ul> </li> <li>Cycle Removal Options:               <ul style="list-style-type: none"> <li>IsCycleRemoveGreedy</li> <li>IsCycleRemoveDFS.</li> </ul> </li> <li>Layering Options:               <ul style="list-style-type: none"> <li>IsLayeringLongestPathSink</li> <li>IsLayeringLongestPathSource</li> <li>IsLayeringOptimalLinkLength.</li> </ul> </li> <li>Initialize Options:               <ul style="list-style-type: none"> <li>IsInitializeNaive</li> <li>IsInitializeDFSOut</li> <li>IsInitializeDFSIn.</li> </ul> </li> <li>Crossing Reduction Option:               <ul style="list-style-type: none"> <li>IsCrossReduceAggressive.</li> </ul> </li> <li>Layout Options - Direction               <ul style="list-style-type: none"> <li>IsLayoutDirectionUp</li> <li>IsLayoutDirectionDown</li> <li>IsLayoutDirectionLeft</li> <li>IsLayoutDirectionRight.</li> </ul> </li> </ul> Parameters: <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to lay out.</li> <li>LayoutStyle: Long - the layout style.</li> <li>Iterations: Long - the number of layout iterations the Layout process should take to perform cross reduction (Default value = 4).</li> <li>LayerSpacing: Long - the per-element layer spacing the Layout process shall use (Default value = 20).</li> <li>ColumnSpacing: Long - the per-element column spacing the Layout process shall use (Default value = 20).</li> <li>SaveToDiagram: Boolean - specifies whether or not Enterprise Architect should save the supplied layout options as default to the</li> </ul> | <a href="#">ConstLayoutStyles Enum</a> <sup>[1832]</sup> |

| Method                                                                  | Type                           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See Also                                                                                                      |
|-------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
|                                                                         |                                | diagram in question.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                               |
| <b>LoadControlledPackage</b><br>(string PackageGUID)                    | String                         | Loads a package that has been marked and configured as controlled. The filename details are stored in the package control data.<br><br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to load.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                               |
| <b>LoadDiagram</b> (string DiagramGUID)                                 | protected abstract:<br>Boolean | Loads a diagram by its GUID.<br><br>Parameter: <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to load; if you retrieve the GUID using the Diagram interface, use the <b>GUIDtoXML</b> function to convert it to XML format.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                      | <a href="#">GUIDtoXML</a>  |
| <b>LoadProject</b> (string FileName)                                    | protected abstract:<br>Boolean | Loads an Enterprise Architect project file<br><br>Do not use this method if you have accessed the Project interface from the Repository, which has already loaded a file<br><br>Parameters: <ul style="list-style-type: none"> <li>FileName: String - the name of the project file to load</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                               |
| <b>Migrate</b> (string GUID, string SourceType, string DestinationType) | Void                           | Migrates a model (or part of a model) from one BPMN or SysML format to an upgraded format<br><br>Parameters: <ul style="list-style-type: none"> <li>GUID: String - the GUID of the package or element for which the contents are to be migrated</li> <li>SourceType: String - the type of model to be upgraded; accepted values: <ul style="list-style-type: none"> <li><b>BPMN</b></li> <li><b>BPMN1.1</b></li> <li><b>SysML1.1</b></li> </ul> </li> <li>DestinationType: String - the type of model to upgrade to; accepted values: <ul style="list-style-type: none"> <li><b>BPMN1.1</b></li> <li><b>BPMN1.1::BPEL</b></li> <li><b>BPMN2.0</b></li> <li><b>SysML1.2</b></li> </ul> </li> </ul> |                                                                                                               |
| <b>MigrateToBPMN11</b><br>(string GUID, string Type)                    | Void                           | Migrates every BPMN 1.0 construct in a package or an element (including elements, attributes, diagrams and connectors) to BPMN 1.1.<br><br>Parameters <ul style="list-style-type: none"> <li>GUID: String - the GUID of the</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                               |

| Method                                                                                      | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See Also                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                             |                | <p>package or element for which the contents are to be migrated to BPMN 1.1</p> <ul style="list-style-type: none"> <li>Type: String - the type of upgrade, either just to BPMN 1.1 or to BPMN 1.1 and BPEL. Accepted values:</li> </ul> <p><b>BPMN</b> = migrate to BPMN 1.1<br/> <b>BPEL</b> = migrate to BPMN 1.1 and update:</p> <ul style="list-style-type: none"> <li>any diagram with stereotype <i>BPMN</i> to <i>BPEL</i></li> <li>any element with stereotype <i>BusinessProcess</i> to <i>BPELProcess</i>.</li> </ul> <p>Migrating to BPEL is possible only in the Ultimate or Business and Software Engineering editions of Enterprise Architect.</p>           |                                                                                                                                                                                              |
| <b>ProjectTransfer</b> (string SourceFilePath, string TargetFilePath, string LogFilePath)   | <i>Boolean</i> | <p>Transfers the project from a .EAP file or DBMS to a .EAP file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>SourceFilePath: String - the path of the source file to transfer.</li> <li>TargetFilePath: String - the path of the target file; Enterprise Architect creates a new Base project in this location.</li> <li>LogFilePath: String - the path of the log file where the status of the transfer process is updated.</li> </ul> <p>In automation, the target file does not have to exist. The file path is enough; Enterprise Architect creates a new, empty Base.EAP file and transfers the source project into it.</p>                        |                                                                                                                                                                                              |
| <b>PublishResult</b> (string CategoryID, EA.EnumMVErrorType ErrorType, string ErrorMessage) | <i>String</i>  | <p>Returns the results of each rule that can be performed during model validation. It must be called once for each rule from the <b>EA_OnInitializeUserRules</b> broadcast handler.</p> <p>The return value is a <i>RuleId</i>, which can be used for reference purposes when an individual rule is executed by Enterprise Architect during model validation.</p> <p>See <b>Model Validation Example</b> for a detailed example of use of this method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>CategoryId: String - should be passed the return value from the <b>DefineRuleCategory</b> method.</li> <li>ErrorType: EA.EnumMVErrorType -</li> </ul> | <p><a href="#">EA_OnInitializeUserRules</a><sup>[2000]</sup></p> <p><a href="#">Model Validation Example</a><sup>[2006]</sup></p> <p><a href="#">DefineRuleCategory</a><sup>[1931]</sup></p> |

| Method                                                                                                           | Type                                  | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also |
|------------------------------------------------------------------------------------------------------------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                                                                  |                                       | <p>depending on the severity of the error being validated, can be:</p> <ul style="list-style-type: none"> <li>• <b>mvErrorCritical</b></li> <li>• <b>mvError</b></li> <li>• <b>mvWarning</b>, or</li> <li>• <b>mvInformation</b>.</li> </ul> <ul style="list-style-type: none"> <li>• ErrorMessage: String - contains an error string.</li> </ul>                                                                                                                                                      |          |
| <b>PutDiagramImageOnClipboard (string DiagramGUID, long Type)</b>                                                | protected abstract:<br><i>Boolean</i> | <p>Copies an image of the specified diagram to the clipboard.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• DiagramGUID: String - the GUID (in XML format) of the diagram to copy.</li> <li>• Type: Long - the file type.</li> <li>• If Type = <b>0</b> then it is a metafile</li> <li>• If Type = <b>1</b> then it is a Device Independent Bitmap.</li> </ul>                                                                                                                       |          |
| <b>PutDiagramImageToFile (string Diagram GUID, string FileName, long Type)</b>                                   | protected abstract:<br><i>Boolean</i> | <p>Saves an image of the specified diagram to file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• DiagramGUID: String - the GUID (in XML format) of the diagram to save.</li> <li>• FileName: String - the name of the file to save the diagram into.</li> <li>• Type: Long - the file type.</li> <li>• If type = <b>0</b> then it is a metafile</li> <li>• If type = <b>1</b> then it uses the file type from the name extension (that is, .bmp, .jpg, .gif, .png, .tga)</li> </ul> |          |
| <b>ReloadProject ()</b>                                                                                          | protected abstract:<br><i>Boolean</i> | <p>Reloads the current project. This is a convenient method to refresh the current loaded project (in case of outside changes to the .EAP file).</p>                                                                                                                                                                                                                                                                                                                                                   |          |
| <b>RunReport (string PackageGUID, string TemplateName, string Filename)</b>                                      | protected abstract:<br><i>Void</i>    | <p>Runs a named RTF report.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package to run the report on.</li> <li>• TemplateName: String - the RTF report template to use. If the PackageGUID has a stereotype of <i>MasterDocument</i>, the template is not required.</li> <li>• FileName: String - the file name to store the generated report in.</li> </ul>                                                                 |          |
| <b>RunHTMLReport (string PackageGUID, string ExportPath, string ImageFormat, string Style, string Extension)</b> | <i>String</i>                         | <p>Runs an HTML report (same as <b>Documentation   HTML Documentation</b> when you right-click on a package in the <b>Project Browser</b>).</p>                                                                                                                                                                                                                                                                                                                                                        |          |

| Method                                                              | Type                                 | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                | See Also |
|---------------------------------------------------------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                                                                     |                                      | Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to run the report on.</li> <li>ExportPath: String - the file name to store the generated report in.</li> <li>ImageFormat: String.</li> <li>Style: String.</li> <li>Extension: String.</li> </ul>                                                                                                                    |          |
| <b>SaveControlledPackage (string PackageGUID)</b>                   | <i>String</i>                        | Saves a package that has been configured as a controlled package, to XML. Only the package GUID is required, Enterprise Architect picks the rest up from the package control information.<br>Parameter: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to save.</li> </ul>                                                                                                     |          |
| <b>SaveDiagramImageToFile (string Filename)</b>                     | protected abstract:<br><i>String</i> | Saves a diagram image of the current diagram to file.<br>Parameters: <ul style="list-style-type: none"> <li>FileName: String - the filename of the image to save.</li> </ul>                                                                                                                                                                                                                                                         |          |
| <b>ShowWindow (long Show)</b>                                       | protected abstract:<br><i>Void</i>   | Shows or hides the Enterprise Architect User Interface.<br>Parameters: <ul style="list-style-type: none"> <li>Show: Long.</li> </ul>                                                                                                                                                                                                                                                                                                 |          |
| <b>SynchronizeClass (string ElementGUID, string ExtraOptions)</b>   | <i>Boolean</i>                       | Synchronizes a Class with the latest source code.<br>Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to update from code.</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently unused.</li> </ul>                                                                                                                            |          |
| <b>SynchronizePackage (string PackageGUID, string ExtraOptions)</b> | <i>Boolean</i>                       | Synchronizes each Class in a package with the latest source code.<br>Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package containing the elements to update from code.</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently enables synchronization of all child packages (children) - for example: <i>children=1</i>.</li> </ul> |          |

| Method                                                                                                                      | Type           | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See Also |
|-----------------------------------------------------------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>TransformElement</b><br>(string TransformName,<br>string ElementGUID,<br>string TargetPackage,<br>string ExtraOptions)   | <i>Boolean</i> | Transforms an element into a package.<br>Parameters: <ul style="list-style-type: none"> <li>• TransformName: String - specifies the transformation that should be executed.</li> <li>• ElementGUID: String - the GUID (in XML format) of the element to transform.</li> <li>• TargetPackageGUID: String - the GUID (in XML format) of the package to transform into.</li> <li>• ExtraOptions: String - enables extra options to be given to the command: <ul style="list-style-type: none"> <li>• GenCode=True/False - articulate code generation from the transformed elements; this option supercedes the current model setting.</li> </ul> </li> </ul>                                                                                                                                         |          |
| <b>TransformPackage</b><br>(string TransformName,<br>string SourcePackage,<br>string TargetPackage,<br>string ExtraOptions) | <i>Boolean</i> | Runs a transformation on the contents of a package.<br>Parameters: <ul style="list-style-type: none"> <li>• TransformName: String - specifies the transformation that should be executed.</li> <li>• SourcePackageGUID: String - the GUID (in XML format) of the package to transform.</li> <li>• TargetPackageGUID: String - the GUID (in XML format) of the package to transform into.</li> <li>• ExtraOptions: String - enables extra options to be given to the command: <ul style="list-style-type: none"> <li>• GenCode=true/false - articulate code generation from the transformed elements; this option supercedes the current model setting.</li> <li>• SubPackages=true/false - specify if the child packages are to be included whilst transforming a package.</li> </ul> </li> </ul> |          |
| <b>XMLtoGUID</b> (string GUID)                                                                                              | <i>String</i>  | Changes a GUID in XML format to the form used inside Enterprise Architect.<br>Parameters: <ul style="list-style-type: none"> <li>• GUID: String - the XML style GUID to convert to Enterprise Architect internal format.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |

Notes:

- These methods all require input GUIDs in XML format; use **GUIDtoXML** to change the Enterprise Architect GUID to an XML GUID

**Learn More:**

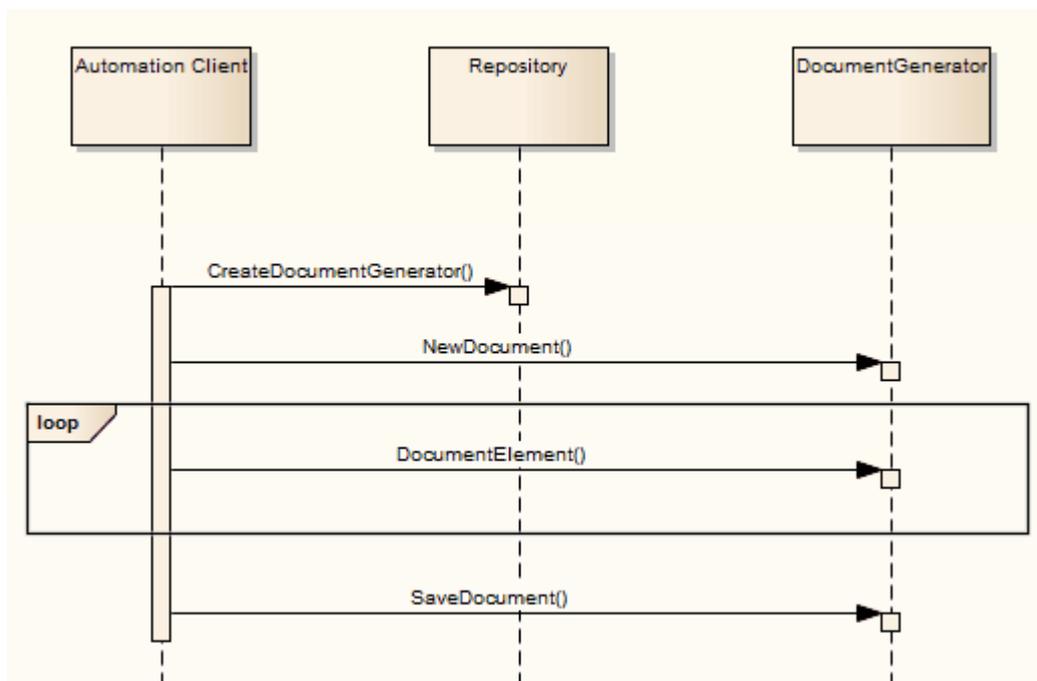
- [GUIDtoXML](#) [1939]

### 18.2.2.10 Document Generator Interface Package

The *Enterprise Architect.DocumentGenerator* interface. This is the XML-based interface to Enterprise Architect RTF and HTML reporting. You can get a pointer to this interface using the *Repository.CreateDocumentGenerator* method.

The following diagram illustrates how you might apply this interface in generating a report through the Automation Interface.

**Example:**



#### 18.2.2.10.1 DocumentGenerator Class

The *DocumentGenerator* Class can be accessed from the *Repository* using *CreateProjectGenerator()*. The returned interface provides access to the XML-based Enterprise Architect Automation Interface. Use this interface to get XML for the various internal elements and to run some utility functions to perform tasks such as load diagrams or run reports.

**DocumentGenerator Attributes:**

| Attribute         | Type                                             | Notes                                                                     |
|-------------------|--------------------------------------------------|---------------------------------------------------------------------------|
| <b>ObjectType</b> | <a href="#">ObjectType</a> <small>[1835]</small> | Read only. Distinguishes objects referenced through a Dispatch interface. |

Document Generator Methods:

| Method                                                                       | Type           | Notes                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DocumentDiagram (long DiagramID, long nDepth, string templateName)</b>    | <i>Boolean</i> | Documents a diagram.<br>Parameters: <ul style="list-style-type: none"> <li>DiagramId: Long - the ID of the diagram.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting diagrams; can be blank.</li> </ul>                      |
| <b>DocumentElement (long ElementID, long nDepth, string templateName)</b>    | <i>Boolean</i> | Documents an element.<br>Parameters: <ul style="list-style-type: none"> <li>ElementId: Long - the ID of the element.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting elements; can be blank.</li> </ul>                     |
| <b>DocumentModelAuthor (string name, long nDepth, string templateName)</b>   | <i>Boolean</i> | Documents a model author.<br>Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the author.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting model authors; can be blank.</li> </ul>              |
| <b>DocumentModelClient (string name, long nDepth, string templateName)</b>   | <i>Boolean</i> | Documents a single model client.<br>Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the client.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting model clients; can be blank.</li> </ul>       |
| <b>DocumentModelGlossary (long id, long nDepth, string templateName)</b>     | <i>Boolean</i> | Documents a single model glossary term.<br>Parameters: <ul style="list-style-type: none"> <li>id: Long - the ID of the term.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting model glossary terms; can be blank.</li> </ul> |
| <b>DocumentModelIssue (long id, long nDepth, string templateName)</b>        | <i>Boolean</i> | Documents a single model issue.<br>Parameters: <ul style="list-style-type: none"> <li>id: Long - the ID of the issue.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting model issues; can be blank.</li> </ul>                |
| <b>DocumentModelResource (string name, long nDepth, string templateName)</b> | <i>Boolean</i> | Documents a single model resource.<br>Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the resource.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting model resources; can be blank.</li> </ul> |

| Method                                                                    | Type           | Notes                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DocumentModelRole</b> (string name, long nDepth, string templateName)  | <i>Boolean</i> | Documents a single model role.<br>Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the role.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting model roles; can be blank.</li> </ul>                                                             |
| <b>DocumentModelTask</b> (long id, long nDepth, string templateName)      | <i>Boolean</i> | Documents a single model task.<br>Parameters: <ul style="list-style-type: none"> <li>id: Long - the ID of the the task.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting model tasks; can be blank.</li> </ul>                                                               |
| <b>DocumentPackage</b> (long PackageID, long nDepth, string templateName) | <i>Boolean</i> | Documents a package.<br>Parameters: <ul style="list-style-type: none"> <li>PackageId: Long - the ID of the package.</li> <li>nDepth: Long - adjust the heading level by nDepth.</li> <li>templateName: String - a template to use when documenting packages; can be blank.</li> </ul>                                                                      |
| <b>GetLastError()</b>                                                     | <i>String</i>  | Returns a string value describing the most recent error that occurred in relation to this object.                                                                                                                                                                                                                                                          |
| <b>InsertBreak</b> (enum breakType)                                       | <i>Boolean</i> | Inserts a break into the report at the current location.<br>Parameters: <ul style="list-style-type: none"> <li>breakType: enum - <b>0</b> = page break, <b>1</b> = section break.</li> </ul>                                                                                                                                                               |
| <b>InsertLinkedDocument</b> (string guid)                                 | <i>Boolean</i> | Inserts a linked document into the report at the current location.<br><br>A linked document can used to set the header and footer of the report. These are taken from the first linked document added to the report.<br>Parameters: <ul style="list-style-type: none"> <li>guid: String - the GUID of the element that has a linked document.</li> </ul>   |
| <b>InsertTableOfContents</b>                                              | <i>Boolean</i> | Inserts a Table of Contents at the current position.                                                                                                                                                                                                                                                                                                       |
| <b>InsertTeamReviewPost</b> (string path)                                 | <i>Boolean</i> | Inserts a Team Review posting into the report at the current location.<br>Parameters: <ul style="list-style-type: none"> <li>path: String - the path of the Team Review post.</li> </ul>                                                                                                                                                                   |
| <b>InsertText</b> (string Text, string style)                             | <i>Boolean</i> | Inserts some static text into the report at the current location.<br><br>No carriage return is added, so remember to add one if required.<br>Parameters: <ul style="list-style-type: none"> <li>Text: String - the static text to be inserted.</li> <li>Style: String - the name of the style in the template; defaults to <i>Normal</i> style.</li> </ul> |

| Method                                                    | Type           | Notes                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NewDocument (string templateName)</b>                  | <i>Boolean</i> | Starts a new document; must be called before attempting to document anything else.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>templateName: String - a template to use when documenting elements. Can be blank.</li> </ul>                                                                                                     |
| <b>ReplaceField (string fieldname, string fieldvalue)</b> | <i>Boolean</i> | Updates the fieldname with the value given.<br><br>If you call this function more than once with the same fieldname, the field only has the last value set.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>fieldname: String - the fieldname to find.</li> <li>fieldvalue: String - the value to insert into the field.</li> </ul> |
| <b>SaveDocument (string filename, enum nDocType)</b>      | <i>Boolean</i> | Saves the document to disk.<br><br>Parameters:<br><ul style="list-style-type: none"> <li>filename: String - the filename to save the file to.</li> <li>nDocType: Enum - <b>0</b> = RTF, <b>1</b> = HTML.</li> </ul>                                                                                                                              |

### 18.2.2.11 Mail Interface Package

The Enterprise Architect *MailInterface* package contains:

- A function to retrieve a pointer to the interface
- Functions to create and send a mail message within the current mode
- Utility functions for creating hyperlinks to selected model elements

You can get a pointer to this interface using the method **Repository.GetMailInterface**.

#### Learn more:

- [Repository Package](#) <sup>[1838]</sup>
- [MailInterface Class](#) <sup>[1950]</sup>

#### 18.2.2.11.1 MailInterface Class

The *MailInterface* interface can be accessed from the Repository using **GetMailInterface()**. The returned interface provides access to the Enterprise Architect Model Mail Interface. Use this interface to automate the process of creating and sending messages using Enterprise Architect's Model Mail system.

#### MailInterface Attributes:

| Attribute               | Type                                             | Notes                                                                          |
|-------------------------|--------------------------------------------------|--------------------------------------------------------------------------------|
| <b>ObjectType</b>       | <a href="#">ObjectTyp</a><br><sup>e</sup> [1835] | Read Only<br><br>Distinguishes objects referenced through a dispatch interface |
| <b>MessagingEnabled</b> | Boolean                                          | Read Only<br><br>Advises whether messaging is enabled on the current model     |

#### MailInterface Methods:

| Method                                                               | Type          | Notes                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------------------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GetLastError ( )</b>                                              | <i>String</i> | Returns the last error message set for the MailInterface                                                                                                                                                                                                                                                                                                     |
| <b>GetRecipientGUID (string UserName)</b>                            | <i>String</i> | Returns the GUID of the specified Enterprise Architect user<br>Parameters: <ul style="list-style-type: none"> <li>• UserName: String - The name of a user defined in the current model</li> </ul>                                                                                                                                                            |
| <b>GetElementHyperlink (string ElementGUID, string LinkText)</b>     | <i>String</i> | Returns a string containing a hyperlink to the element specified by the input parameter ElementGUID<br>Parameters: <ul style="list-style-type: none"> <li>• ElementGUID: String - The GUID of the element for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the element name)</li> </ul>           |
| <b>GetAttributeHyperlink (string AttributeGUID, string LinkText)</b> | <i>String</i> | Returns a string containing a hyperlink to the attribute specified by the input parameter AttributeGUID<br>Parameters: <ul style="list-style-type: none"> <li>• AttributeGUID: String - The GUID of the attribute for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the attribute name)</li> </ul> |
| <b>GetMethodHyperlink (string MethodGUID, string LinkText)</b>       | <i>String</i> | Returns a string containing a hyperlink to the method specified by the input parameter MethodGUID<br>Parameters: <ul style="list-style-type: none"> <li>• MethodGUID: String - The GUID of the method for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the method name)</li> </ul>                |
| <b>GetDiagramHyperlink (string DiagramGUID, string LinkText)</b>     | <i>String</i> | Returns a string containing a hyperlink to the diagram specified by the input parameter DiagramGUID<br>Parameters: <ul style="list-style-type: none"> <li>• DiagramGUID: String - The GUID of the diagram for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the diagram name)</li> </ul>           |
| <b>GetPackageHyperlink (string PackageGUID, string LinkText)</b>     | <i>String</i> | Returns a string containing a hyperlink to the package specified by the input parameter PackageGUID<br>Parameters: <ul style="list-style-type: none"> <li>• PackageGUID: String - The GUID of the package for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the package name)</li> </ul>           |
| <b>GetFileHyperlink (string FilePath, string LinkText)</b>           | <i>String</i> | Returns a string containing a hyperlink to the file specified by the input parameter FilePath<br>Parameters: <ul style="list-style-type: none"> <li>• FilePath: String - The path name of the file for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. The</li> </ul>                                |

| Method                                                                                                                                | Type    | Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                       |         | file's name)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>GetWebHyperlink</b><br>(string URL, string LinkText)                                                                               | String  | Returns a string containing a hyperlink to the URL specified by the input parameter URL<br><br>Parameters: <ul style="list-style-type: none"> <li>• URL: String - The URL of the item for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>ComposeMailMessage</b><br>(string InitialRecipientGUID, string InitialSubject, messageflag InitialFlag, string InitialMessageText) | Boolean | Creates a new mail message using the values specified in the input parameters; the message is displayed in the composition window, ready for sending<br><br>This method does NOT send the message<br><br>Parameters: <ul style="list-style-type: none"> <li>• InitialRecipientGUID: String - Initial value for the GUID of the addressee user (an Enterprise Architect user defined in the current model)</li> <li>• InitialSubject: String - Initial value for the Subject text to display for this message</li> <li>• InitialFlag: MessageFlag - Initial value for the flag type/color to attach to this message</li> <li>• InitialMessageText: String - Initial value for the text that is the body of the message</li> </ul> |
| <b>SendMailMessage</b><br>(string RecipientGUID, string Subject, messageflag Flag, string MessageText)                                | Boolean | Creates and sends a new mail message using the values specified in the input parameters<br><br>Parameters: <ul style="list-style-type: none"> <li>• RecipientGUID: String - The GUID of the addressee user (an Enterprise Architect user defined in the current model)</li> <li>• Subject: String - The Subject text to display for this message</li> <li>• Flag: MessageFlag - The flag type/color to attach to this message</li> <li>• MessageText: String - The text that is the body of the message</li> </ul>                                                                                                                                                                                                               |

### 18.2.2.12 Code Samples

This topic contains various code examples indicating how to use the Automation Interface, written in VB.Net:

| Topic                       | Link                                                          |
|-----------------------------|---------------------------------------------------------------|
| Open the Repository         | <a href="#">Open the Repository</a> <sup>[1953]</sup>         |
| Iterate Through a .EAP File | <a href="#">Iterate Through a .EAP File</a> <sup>[1953]</sup> |
| Add and Manage Packages     | <a href="#">Add and Manage Packages</a> <sup>[1954]</sup>     |
| Add and Manage Elements     | <a href="#">Add and Manage Elements</a> <sup>[1955]</sup>     |
| Add a Connector             | <a href="#">Add a Connector</a> <sup>[1956]</sup>             |
| Add and Manage Diagrams     | <a href="#">Add and Manage Diagrams</a> <sup>[1957]</sup>     |
| Add and Delete Features     | <a href="#">Add and Delete Features</a> <sup>[1957]</sup>     |
| Element Extras              | <a href="#">Element Extras</a> <sup>[1958]</sup>              |

| Topic                | Link                                                   |
|----------------------|--------------------------------------------------------|
| Repository Extras    | <a href="#">Repository Extras</a> <sup>[1961]</sup>    |
| Stereotypes          | <a href="#">Stereotypes</a> <sup>[1963]</sup>          |
| Work with Attributes | <a href="#">Work with Attributes</a> <sup>[1964]</sup> |
| Work with Methods    | <a href="#">Work with Methods</a> <sup>[1965]</sup>    |

### 18.2.2.12.1 Open the Repository

#### Topics:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' An example of how to open an Enterprise Architect repository '' in VB.Net.  Public Class AutomationExample      '' class level variable for Repository     Public m_Repository As Object      Public Sub Run()         try             '' create the repository object             m_Repository = CreateObject("EA.Repository")              '' open an EAP file             m_Repository.OpenFile("F:\Test\EAAuto.EAP")             '' use the Repository in any way required             DumpModel              '' close the repository and tidy up             m_Repository.Exit()             m_Repository = Nothing              .... catch e as exception             Console.WriteLine(e)         End try     End Sub end Class </pre> |

### 18.2.2.12.2 Iterate Through a .EAP File

#### Topics:

| Topic                | Detail                                                                                                                                                                                                                                                      |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' Assume repository has already been opened.  '' Start at the model level Sub DumpModel()     Dim idx as Integer     For idx=0 to m_Repository.Models.Count - 1         DumpPackage("", m_Repository.Models.Get At (idx))     End For End Sub </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> Next End Sub  ' output package name, then element contents, then process child packages Sub DumpPackage(Indent as String, Package as Object)     Dim idx as Integer     Console.WriteLine(Indent + Package.Name)     DumpElements(Indent + "    ", Package)      For idx = 0 to Package.Packages.Count - 1         DumpPackage(Indent + "    ", Package. Packages. Get At (idx))     Next End Sub  '' dump element name Sub DumpElements(Indent as String, Package as Object)     Dim idx as Integer     For idx = 0 to Package.Elements.Count - 1         Console.WriteLine(Indent + "::" + Package. Elements. Get At (idx). Name)     Next End Sub </pre> |

### 18.2.2.12.3 Add and Manage Packages

#### Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| public Object | <pre> Example illustrating how to add a Model or a Package.  Sub TestPackageLifecycle      Dim idx as integer     Dim idx2 as integer     Dim package as object     Dim model as object     Dim o as object      '' first add a new Model      model = m_Repository.Models.AddNew ("AdvancedModel", "")     If not model.Update() Then         Console.WriteLine(model.LastError())     End If      '' refresh the models collection     m_Repository.Models.Refresh      '' now work through models collection and add a package      For idx = 0 to m_Repository.Models.Count - 1         o = m_Repository.Models.Get At (idx)         Console.WriteLine(o.Name)         If o.Name = "AdvancedModel" Then             package = o.Packages.AddNew ("Subpackage", "Nothing")             If not package.Update() Then                 Console.WriteLine(package. </pre> |

| Topic | Detail                                                                                                                                                                                                                                                      |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> Get Last Error ()     End If      package.Element.Stereotype = "system"     package.Update      ''for testing purposes just delete the     ''newly created Model and its contents     m_Repository.Models.Delete(idx)      End If Next End Sub </pre> |

#### 18.2.2.12.4 Add and Manage Elements

##### Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| public Object | <pre> '' Add and delete elements in a package.  Sub ElementLifecycle      Dim package as Object     Dim element as Object      package = m_Repository.GetPackageById(2)     element = package.elements.AddNew("Login to Website", "UseCase")     element.Stereotype = "testcase"     element.Update     package.elements.Refresh()      Dim idx as integer      ''note the repeated calls to "package.elements. Get At "     ''in general you should make this call once and assign to a local     ''variable - in the example below, Enterprise Architect loads the element required     ''everytime a call is made - rather than loading once and keeping     ''a local reference      For idx = 0 to package.elements.count - 1         Console.WriteLine(package.elements.Get At (idx) . Name)         If (package.elements.Get At (idx) . Name = "Login to Website" and _             package.elements.Get At (idx) . Type = "UseCase") Then             package.elements.deleteat (idx, false)         End If     Next End Sub </pre> |

## 18.2.2.12.5 Add a Connector

Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| public Object | <pre> " Add a connector and set values.  Sub ConnectorTest      Dim source as object     Dim target as object     Dim con as object     Dim o as object      Dim client as object     Dim supplier as object      '' use ElementID's to quickly load an element in this example     ''... you must find suitable ID's in your model      source = m_Repository.GetElementByID(129)     target = m_Repository.GetElementByID(169)      con = source.Connectors.AddNew("test link 2", "Association")      '' again- replace ID with a suitable one from your model     con.SupplierID = 169      If not con.Update Then         Console.WriteLine(con.GetLastError)     End If     source.Connectors.Refresh      Console.WriteLine("Connector Created")      o = con.Constraints.AddNew("constraint 2","type")     If not o.Update Then         Console.WriteLine(o.GetLastError)     End If      o = con.TaggedValues.AddNew("Tag","Value")     If not o.Update Then         Console.WriteLine(o.GetLastError)     End If      '' use the client and supplier ends to set '' additional information      client = con.ClientEnd     client.Visibility = "Private"     client.Role = "m_client"     client.Update     supplier = con.SupplierEnd     supplier.Visibility = "Protected"     supplier.Role = "m_supplier"     supplier.Update      Console.WriteLine("Client and Supplier set")      Console.WriteLine(client.Role)     Console.WriteLine(supplier.Role)  End Sub </pre> |

**18.2.2.12.6 Add and Manage Diagrams**Topics:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> '' An example of how to create a diagram and add an element to it. '' Note the optional use of element rectangle setting using '' left, right, top and bottom dimensions in AddNew call.  Sub DiagramLifeCycle      Dim diagram as object     Dim v as object     Dim o as object     Dim package as object      Dim idx as Integer     Dim idx2 as integer      package = m_Repository.GetPackageById(5)      diagram = package.Diagrams.AddNew("Logical Diagram", "Logical")     If not diagram.Update Then         Console.WriteLine(diagram.GetLastError)     End If      diagram.Notes = "Hello there this is a test"     diagram.update()      o = package.Elements.AddNew ("ReferenceType", "Class")     o.Update      '' add element to diagram - supply optional rectangle co-ordinates      v = diagram.DiagramObjects.AddNew("l=200;r=400; t=200;b=600;", "")     v.ElementID = o.ElementID     v.Update      Console.WriteLine(diagram.DiagramID)  End Sub </pre> |

**18.2.2.12.7 Add and Delete Features**Topics:

| Topic                | Detail                                                                                                                                                                                                                       |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>public Object</b> | <pre> Dim element as object Dim idx as integer Dim attribute as object Dim method as object  'just load an element by ID - you must 'substitute a valid ID from your model element = m_Repository.GetElementById(246) </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> '' create a new method method = element.Methods.AddNew("newMethod", "int") method.Update element.Methods.Refresh  'now loop through methods for Element - and delete our addition For idx = 0 to element.Methods.Count - 1     method =element.Methods.GetAt(idx)     Console.WriteLine(method.Name)     If(method.Name = "newMethod") Then         element.Methods.Delete(idx)     End if Next  ' create an attribute attribute = element.attributes.AddNew("NewAttribute", "int") attribute.Update element.attributes.Refresh  'loop through and delete our new attribute For idx = 0 to element.attributes.Count - 1     attribute =element.attributes.GetAt(idx)     Console.WriteLine(attribute.Name)     If(attribute.Name = "NewAttribute") Then         element.attributes.Delete(idx)     End If Next </pre> |

### 18.2.2.12.8 Element Extras

#### Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| public Object | <pre> '' Examples of how to access and use element extras, such as '' scenarios, constraints and requirements.  Sub ElementExtras      Dim element as object     Dim o as object     Dim idx as Integer     Dim bDel as boolean     bDel = true      try         element = m_Repository.GetElementById(129)          'manage constraints for an element         'demonstrate addnew and delete         o = element.Constraints.AddNew ("Appended", "Type")         If not o.Update Then             Console.WriteLine("Constraint error:" + o. GetLastError())         End if         element.Constraints.Refresh         For idx = 0 to element.Constraints.Count - 1             o = element.Constraints.GetAt(idx)             Console.WriteLine(o.Name) </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> If (o.Name="Appended") Then   If bDel Then element.Constraints.Delete (i dx) End if Next  ' efforts o = element.Efforts.AddNew("Appended", "Type") If not o.Update Then   Console.WriteLine("Efforts error:" + o. Get Last Error()) End if element.Efforts.Refresh For idx = 0 to element.Efforts.Count - 1   o = element.Efforts.Get At (idx)   Console.WriteLine(o.Name)   If (o.Name="Appended") Then     If bDel Then element.Efforts.Delete (i dx)   End if Next  ' Ri sks o = element.Ri sks.AddNew("Appended", "Type") If not o.Update Then   Console.WriteLine("Ri sks error:" + o. Get Last Error()) End if element.Ri sks.Refresh For idx = 0 to element.Ri sks.Count - 1   o = element.Ri sks.Get At (idx)   Console.WriteLine(o.Name)   If (o.Name="Appended") Then     If bDel Then element.Ri sks.Delete (idx)   End if Next  ' Metrics o = element.Metrics.AddNew("Appended", "Change") If not o.Update Then   Console.WriteLine("Metrics error:" + o. Get Last Error()) End if element.Metrics.Refresh For idx = 0 to element.Metrics.Count - 1   o = element.Metrics.Get At (idx)   Console.WriteLine(o.Name)   If (o.Name="Appended") Then     If bDel Then element.Metrics.Delete (i dx)   End if Next  ' TaggedVal ues o = element.TaggedVal ues.AddNew ("Appended", "Change") If not o.Update Then   Console.WriteLine("TaggedVal ues error:" + o. Get Last Error()) End if element.TaggedVal ues.Refresh For idx = 0 to element.TaggedVal ues.Count - 1   o = element.TaggedVal ues.Get At (idx)   Console.WriteLine(o.Name)   If (o.Name="Appended") Then </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre>                 If bDel Then element.TaggedValues. Delete (idx)                 End if             Next              ' Scenarios             o = element.Scenarios.AddNew ("Appended", "Change")             If not o.Update Then                 Console.WriteLine("Scenarios error:" + o. GetLastError())             End if             element.Scenarios.Refresh             For idx = 0 to element.Scenarios.Count - 1                 o = element.Scenarios.GetAt(idx)                 Console.WriteLine(o.Name)                 If(o.Name="Appended") Then                     If bDel Then element.Scenarios.Delete (idx)                 End if             Next              ' Files             o = element.Files.AddNew("MyFile", "doc")             If not o.Update Then                 Console.WriteLine("Files error:" + o. GetLastError())             End if             element.Files.Refresh             For idx = 0 to element.Files.Count - 1                 o = element.Files.GetAt(idx)                 Console.WriteLine(o.Name)                 If(o.Name="MyFile") Then                     If bDel Then element.Files.Delete (idx)                 End if             Next              ' Tests             o = element.Tests.AddNew("Test Plan", "Load")             If not o.Update Then                 Console.WriteLine("Tests error:" + o. GetLastError())             End if             element.Tests.Refresh             For idx = 0 to element.Tests.Count - 1                 o = element.Tests.GetAt(idx)                 Console.WriteLine(o.Name)                 If(o.Name="Test Plan") Then                     If bDel Then element.Tests.Delete (idx)                 End if             Next              ' Defect             o = element.Issues.AddNew("Broken", "Defect")             If not o.Update Then                 Console.WriteLine("Issues error:" + o. GetLastError())             End if             element.Issues.Refresh             For idx = 0 to element.Issues.Count - 1                 o = element.Issues.GetAt(idx)                 Console.WriteLine(o.Name)                 If(o.Name="Broken") Then                     If bDel Then element.Issues.Delete (idx)                 End if             End if </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> Next     ' Change     o = element.Issues.AddNew("Change", "Change")     If not o.Update Then         Console.WriteLine("Issues error:" + o. GetLastError())     End if     element.Issues.Refresh     For idx = 0 to element.Issues.Count - 1         o = element.Issues.GetAt(idx)         Console.WriteLine(o.Name)         If(o.Name="Change") Then             If bDel Then element.Issues.Delete (idx)         End if     Next Next  catch e as exception Console.WriteLine(element.Methods.GetLastError ()) Console.WriteLine(e) End try  End Sub </pre> |

### 18.2.2.12.9 Repository Extras

#### Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| public Object | <pre> '' Examples of how to access repository '' collections for system level information.  Sub RepositoryExtras      Dim o as object     Dim idx as integer      ' issues     o = m_Repository.Issues.AddNew("Problem", "Type")     If(o.Update=false) Then         Console.WriteLine(o.GetLastError())     End if     o = nothing     m_Repository.Issues.Refresh     For idx = 0 to m_Repository.Issues.Count - 1         Console.WriteLine(m_Repository.Issues.GetAt (idx).Name)         If(m_Repository.Issues.GetAt(idx).Name = "Problem") then             m_Repository.Issues.DeleteAt(idx, false)             Console.WriteLine("Delete Issues")         End if     Next Next  '' tasks o = m_Repository.Tasks.AddNew("Task 1", "Task type") If(o.Update=false) Then     Console.WriteLine("error - " + o.GetLastError ()) End if </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> o = nothing m_Repository.Tasks.Refresh For idx = 0 to m_Repository.Tasks.Count - 1   Console.WriteLine(m_Repository.Tasks.Get At (idx).Name)   If(m_Repository.Tasks.Get At (idx).Name = "Task 1") then     m_Repository.Tasks.DeleteAt (idx, false)     Console.WriteLine("Delete Tasks")   End if Next  ' glossary o = m_Repository.Terms.AddNew("Term 1", "business") If(o.Update=false) Then   Console.WriteLine ("error - " + o.GetLastError ()) End if o = nothing m_Repository.Terms.Refresh For idx = 0 to m_Repository.Terms.Count - 1   Console.WriteLine(m_Repository.Terms.Get At (idx).Term)   If(m_Repository.Terms.Get At (idx).Term = "Term 1") then     m_Repository.Terms.DeleteAt (idx, false)     Console.WriteLine("Delete Terms")   End if Next  ' authors o = m_Repository.Authors.AddNew("Joe B", "Writer") If(o.Update=false) Then   Console.WriteLine (o.GetLastError()) End if o = nothing m_Repository.Authors.Refresh For idx = 0 to m_Repository.authors.Count - 1   Console.WriteLine(m_Repository.Authors.Get At (idx).Name)   If(m_Repository.authors.Get At (idx).Name = "Joe B") then     m_Repository.authors.DeleteAt (idx, false)     Console.WriteLine("Delete Authors")   End if Next  o = m_Repository.Clients.AddNew("Joe Sphere", "Client") If(o.Update=false) Then   Console.WriteLine (o.GetLastError()) End if o = nothing m_Repository.Clients.Refresh For idx = 0 to m_Repository.Clients.Count - 1   Console.WriteLine(m_Repository.Clients.Get At (idx).Name)   If(m_Repository.Clients.Get At (idx).Name = "Joe Sphere") then     m_Repository.Clients.DeleteAt (idx, false)     Console.WriteLine("Delete Clients")   End if Next  o = m_Repository.Resources.AddNew("Joe </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> Worker", "Resource") If (o.Update=false) Then     Console.WriteLine(o.GetLastError()) End if o = nothing m_Repository.Resources.Refresh For idx = 0 to m_Repository.Resources.Count - 1     Console.WriteLine(m_Repository.Resources.GetAt (idx).Name)     If (m_Repository.Resources.GetAt(idx).Name = "Joe Worker") then         m_Repository.Resources.DeleteAt(idx, false)         Console.WriteLine("Delete Resource")     End if Next End Sub </pre> |

### 18.2.2.12.10 Stereotypes

#### Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| public Object | <pre> Sub Test Stereotypes      Dim o as object     Dim idx as integer      '' add a new stereotype to the Stereotypes collection     o = m_Repository.Stereotypes.AddNew ("funky", "class")     If (o.Update=false) Then         Console.WriteLine(o.GetLastError())     End if     o = nothing      '' make sure you refresh m_Repository.Stereotypes.Refresh      '' then iterate through - deleting our new entry in the process     For idx = 0 to m_Repository.Stereotypes.Count - 1         Console.WriteLine(m_Repository.Stereotypes. GetAt(idx).Name)         If (m_Repository.Stereotypes.GetAt(idx).Name = "funky") then             m_Repository.Stereotypes.DeleteAt(idx, false)             Console.WriteLine("Delete element")         End if     Next End Sub </pre> |

## 18.2.2.12.11 Work With Attributes

Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| public Object | <pre> '' An example of working with attributes.  Sub AttributeLifecycle      Dim element as object     Dim o as object     Dim t as object     Dim idx as Integer     Dim idx2 as integer     try         element = m_Repository.GetElementById(129)          For idx = 0 to element.Attributes.Count - 1              Console.WriteLine("attribute=" + element. Attributes.GetAt(idx).Name)              o = element.Attributes.GetAt(idx)             t = o.Constraints.AddNew("&gt; 123", "Precision")             t.Update()             o.Constraints.Refresh             For idx2 = 0 to o.Constraints.Count - 1                 t = o.Constraints.GetAt(idx2)                 Console.WriteLine("Constraint: " + t. Name)                  If(t.Name="&gt; 123") Then                     o.Constraints.DeleteAt(idx2, false)                 End if             Next              For idx2 = 0 to o.TaggedValues.Count - 1                 t = o.TaggedValues.GetAt(idx2)                 If(t.Name = "Type2") Then                     Console.WriteLine("deleting")                     o.TaggedValues.DeleteAt(idx2, true)                 End if             Next              t = o.TaggedValues.AddNew("Type2", "Number")             t.Update             o.TaggedValues.Refresh             For idx2 = 0 to o.TaggedValues.Count - 1                 t = o.TaggedValues.GetAt(idx2)                 Console.WriteLine("Tagged Value: " + t. Name)             Next              If(element.Attributes.GetAt(idx).Name = "m_Tootle") Then                 Console.WriteLine("delete attribute")                 element.Attributes.DeleteAt(idx, false)             End If              Next          catch e as exception             Console.WriteLine(element.Attributes. GetLastError()) </pre> |

| Topic | Detail                                            |
|-------|---------------------------------------------------|
|       | <pre> Console.WriteLine(e) End try End Sub </pre> |

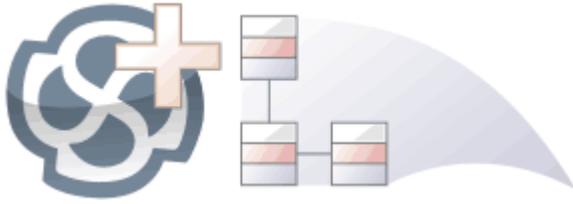
### 18.2.2.12.12 Work With Methods

#### Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| public Object | <pre> '' An example of working with the Methods collection '' of an element - and with Method collections.  Sub MethodLifeCycle      Dim element as object     Dim method as object     Dim t as object     Dim idx as Integer     Dim idx2 as integer      try         element = m_Repository.GetElementByID(129)          For idx = 0 to element.Methods.Count - 1             method = element.Methods.GetAt(idx)             Console.WriteLine(method.Name)              t = method.PreConditions.AddNew ("Test Constraint", "something")             If t.Update = false Then                 Console.WriteLine("PreConditions: " + t.GetLastError)             End If              method.PreConditions.Refresh             For idx2 = 0 to method.PreConditions.Count - 1                 t = method.PreConditions.GetAt(idx2)                 Console.WriteLine("PreConditions: " + t.Name)                 If t.Name = "Test Constraint" Then                     method.PreConditions.DeleteAt(idx2, false)                 End If             Next              t = method.PostConditions.AddNew ("Test Constraint", "something")             If t.Update = false Then                 Console.WriteLine("PostConditions: " + t.GetLastError)             End If              method.PostConditions.Refresh             For idx2 = 0 to method.PostConditions. Count - 1                 t = method.PostConditions.GetAt(idx2)                 Console.WriteLine("PostConditions: " + t.Name)                 If t.Name = "Test Constraint" Then                     method.PostConditions.DeleteAt (idx2, false)                 End If </pre> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <pre> Next     t = method.TaggedValues.AddNew ("Test TaggedValue", "something")     If t.Update = false Then         Console.WriteLine("Tagged Values: " + t.GetLastError)     End If      For idx2 = 0 to method.TaggedValues.Count - 1         t = method.TaggedValues.GetAt(idx2)         Console.WriteLine("Tagged Value: " + t. Name)         If (t.Name= "Test TaggedValue") Then             method.TaggedValues.DeleteAt(idx2, false)         End If     Next      t = method.Parameters.AddNew ("Test Param", "string")     If t.Update = false Then         Console.WriteLine("Parameters: " + t. GetLastError)     End If      method.Parameters.Refresh     For idx2 = 0 to method.Parameters.Count - 1         t = method.Parameters.GetAt(idx2)         Console.WriteLine("Parameter: " + t. Name)         If (t.Name="Test Param") Then             method.Parameters.DeleteAt(idx2, false)         End If     Next      method = nothing Next catch as exception     Console.WriteLine(element.Methods.GetLastError ())     Console.WriteLine(e) End try End Sub </pre> |

## 18.3 Enterprise Architect Add-In Model



### Topics:

| Topic               | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See Also                                                                                                                                                                                                                                                                                |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Introduction</b> | <p>Add-Ins enable you to add functionality to Enterprise Architect. The Enterprise Architect Add-In model builds on the features provided by the <b>Automation Interface</b> to enable you to extend the Enterprise Architect user interface.</p> <p>Add-Ins are ActiveX COM objects that expose public Dispatch methods. They have several advantages over stand-alone automation clients:</p> <ul style="list-style-type: none"> <li>• Add-Ins can define Enterprise Architect menus and sub-menus</li> <li>• Add-Ins receive notifications about various Enterprise Architect user-interface events including menu clicks and file changes</li> <li>• Add-Ins can (and should) be written as in-process (DLL) components. This provides lower call overhead and better integration into the Enterprise Architect environment</li> <li>• Because a current version of Enterprise Architect is already running there is no requirement to start a second copy of Enterprise Architect via the automation interface</li> <li>• Because the Add-In receives object handles associated with the currently running copy of Enterprise Architect, more information is available about the current user's activity; for example, which diagram objects are selected</li> <li>• You are not required to do anything other than to install the Add-In to make it usable; that is, you do not have to configure Add-Ins to run on your systems</li> <li>• Because Enterprise Architect is constantly evolving in response to customer requests, the Add-In interface is flexible:</li> <li>• The Add-In interface does not have its own version, rather it is identified by the version of Enterprise Architect it first appeared in; for example, the current version of the Enterprise Architect Add-In interface is version 2.1</li> <li>• When creating your Add-In, you do not have to subscribe to a type-library</li> </ul> <p>From Enterprise Architect release 7.0 Add-Ins created before 2004 are no longer supported. If an Add-In subscribes to the Addn_Tmpl.tlb interface (2003 style), it fails on load. In this event, contact the vendor or author of the Add-In and request an upgrade.</p> <ul style="list-style-type: none"> <li>• Add-Ins do not have to implement methods that they never use</li> <li>• Add-Ins prompt users via context menus in the tree view and the diagram</li> <li>• Menu check and disable states can be controlled by the Add-In</li> </ul> <p>Add-Ins enhance the existing functionality of Enterprise Architect through a variety of mechanisms such as <b>Scripts</b>, <b>UML Profiles</b> and the <b>Automation Interface</b>. Once an <b>Add-In</b> is registered, it can be</p> | <p><a href="#">Automation Interface</a> <sup>[1822]</sup></p> <p><a href="#">Scripts</a> <sup>[1817]</sup></p> <p><a href="#">UML Profiles</a> <sup>[1039]</sup></p> <p><a href="#">Registered Add-in</a> <sup>[2101]</sup></p> <p><a href="#">Add-In Manager</a> <sup>[1974]</sup></p> |

| Topic | Detail                                    | See Also |
|-------|-------------------------------------------|----------|
|       | managed using the <b>Add-In Manager</b> . |          |

### Create and Use Add-Ins

This topic covers the following information on Add-Ins:

| Topic            | Link                                               |
|------------------|----------------------------------------------------|
| Add-In Tasks     | <a href="#">Add-In Tasks</a> <sup>[1968]</sup>     |
| Add-In Events    | <a href="#">Add-In Events</a> <sup>[1976]</sup>    |
| Broadcast Events | <a href="#">Broadcast Events</a> <sup>[1983]</sup> |
| Custom Views     | <a href="#">Custom Views</a> <sup>[2040]</sup>     |
| MDG Add-Ins      | <a href="#">MDG Add-Ins</a> <sup>[2042]</sup>      |

### **18.3.1 Add-In Tasks**

This topic provides instructions on how to create, test, deploy and manage Add-Ins.

| Topic                     | Detail                                               | Link                                                     |
|---------------------------|------------------------------------------------------|----------------------------------------------------------|
| <b>Create an Add-In</b>   |                                                      | <a href="#">Create an Add-In</a> <sup>[1968]</sup>       |
|                           | Define Menu Items                                    | <a href="#">Define Menu Items</a> <sup>[1969]</sup>      |
|                           | Respond to Menu Events                               | <a href="#">Respond to Menu Events</a> <sup>[1980]</sup> |
|                           | Handle Add-In Events                                 | <a href="#">Handle Add-In Events</a> <sup>[1976]</sup>   |
| <b>Deploy your Add-In</b> |                                                      | <a href="#">Deploy your Add-In</a> <sup>[1970]</sup>     |
|                           | Potential Pitfalls                                   | <a href="#">Potential Pitfalls</a> <sup>[1972]</sup>     |
| <b>Manage Add-Ins</b>     |                                                      |                                                          |
|                           | Register an Add-In(developed in-house or brought-in) | <a href="#">Register an Add-In</a> <sup>[2101]</sup>     |
|                           | The Add-In Manager                                   | <a href="#">The Add-In Manager</a> <sup>[1974]</sup>     |

#### **18.3.1.1 Create Add-Ins**

Before you start you must have an application development tool that is capable of creating ActiveX COM objects supporting the IDispatch interface, such as:

- Borland Delphi
- Microsoft Visual Basic
- Microsoft Visual Studio .Net.

You should consider how to **define menu items**. To help with this, you could review some **examples of Automation Interfaces** (this web page provides examples of code used to create Add-Ins for Enterprise Architect).

**How to:**

An Enterprise Architect Add-In can be created in four steps:

| Step | Action                                                                                                                                                         | See also                                                   |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| 1    | Use a development tool to create an ActiveX COM DLL project. Visual Basic users, for example, choose <i>File&gt;Create New Project-ActiveX DLL</i> .           |                                                            |
| 2    | Connect to the interface using the syntax appropriate to the language as detailed in the <b>Connect to the Interface</b> topic.                                | <a href="#">Connect to the Interface</a> <sup>[1823]</sup> |
| 3    | Create a COM Class and implement each of the <b>general Add-In Events</b> applicable to your Add-In. You only have to define methods for events to respond to. | <a href="#">Add-In Events</a> <sup>[1976]</sup>            |
| 4    | Add a registry key that identifies your Add-In to Enterprise Architect, as described in the <b>Deploy Add-Ins</b> topic.                                       | <a href="#">Deploy Add-Ins</a> <sup>[1970]</sup>           |

**Learn More:**

- [Define Menu Items](#) <sup>[1969]</sup>
- [Examples of Automation Interfaces](#) (Online Resource)

**18.3.1.1.1 Define Menu Items****Topics:**

| Topic                      | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Defining Menu Items</b> | <p>Menu items are defined by responding to the <i>GetMenuItems</i> event.</p> <p>The first time this event is called, <i>MenuName</i> is an empty string, representing the top-level menu. For a simple Add-In with just a single menu option you can return a string; for example:</p> <pre>Function EA_GetMenuItems(Repository as EA.Repository, MenuLocation As String, MenuName As String) As Variant     EA_GetMenuItems = "&amp;Joe's Add-In" End Function</pre>                                                                                                                                                                                                                                                                    |
| <b>Defining Sub-Menus</b>  | <p>To define sub-menus, prefix a parent menu with a dash. Parent and sub-items are defined as follows:</p> <pre>Function EA_GetMenuItems(Repository as EA.Repository, MenuLocation As String, MenuName As String) As Variant     Select Case MenuName         Case ""             ' Parent Menu Item             EA_GetMenuItems = "- &amp;Joe's Add-In"         Case "- &amp;Joe's Add-In"             ' Define Sub-Menu Items using the Array notation.             ' In this example, "Diagram" and "Treeview"             compose the "Joe's Add-In" sub-menu.             EA_GetMenuItems = Array("&amp;Diagram", "&amp;Treeview")         Case Else             MsgBox "Invalid Menu", vbCritical     End Select End Function</pre> |

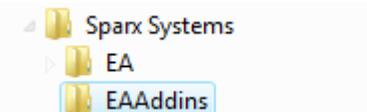
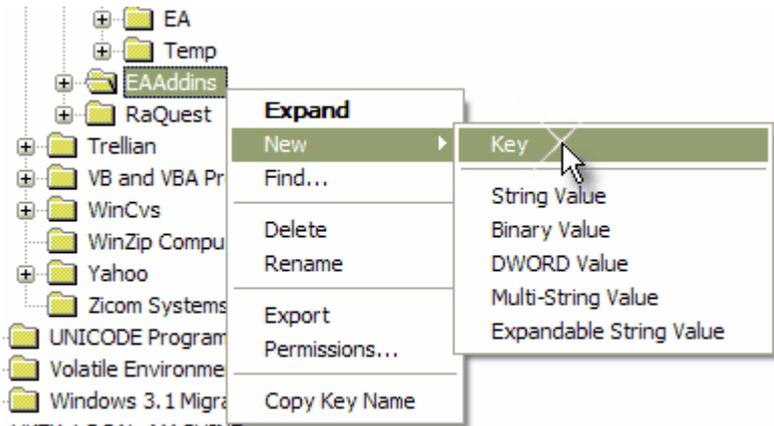
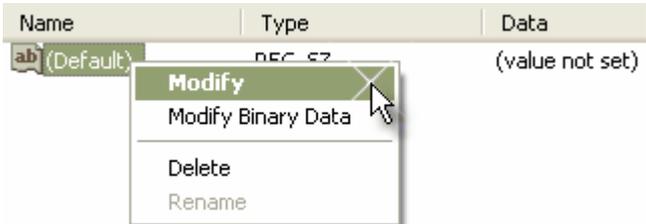
| Topic                                  | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Defining further Sub-Menus</b>      | <p>Similarly, you can define further sub-items:</p> <pre> Function EA_GetMenuItems(Repository As EA.Repository, MenuLocation As String, MenuName As String) As Variant     Select Case MenuName         Case ""             EA_GetMenuItems = "- Joe's Add-In"         Case "- Joe's Add-In"             EA_GetMenuItems = Array("- &amp;Diagram", "&amp;TreeView")         Case "- &amp;Diagram"             EA_GetMenuItems = "&amp;Properties"         Case Else             MsgBox "Invalid Menu", vbCritical         End Select     End Function </pre>                                                                                                                                                                                                                  |
| <b>Enabling/Disabling menu options</b> | <p>To enable or disable menu options by default, you can use this method to show particular items to the user:</p> <pre> Sub EA_GetMenuState(Repository As EA.Repository, Location As String, MenuName As String, ItemName As String, IsEnabled As Boolean, IsChecked As Boolean)     Select Case Location         Case "TreeView"             ' Always enable         Case "Diagram"             ' Always enable         Case "MainMenu"             Select Case ItemName                 Case "&amp;Translate", "Save &amp;Project"                     If GetIsProjectSelected() Then                         IsEnabled = False                     End If             End Select         End Select     End Select     IsChecked = GetIsCurrentSelection() End Sub </pre> |

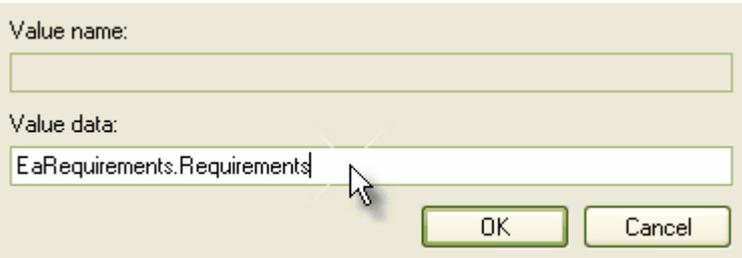
### 18.3.1.1.2 Deploy Add-Ins

#### How to:

To deploy Add-Ins to users' sites, follow the steps below

| Step | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | See also |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1    | Add the Add-In DLL file to an appropriate directory on the user's computer; that is, C:\Program Files\ ( new dir ) .                                                                                                                                                                                                                                                                                                                                                                |          |
| 2    | <p>Register the DLL as appropriate to your platform:</p> <ul style="list-style-type: none"> <li>If compiled as a native Win32 DDL, such as VB6 or C++, register the DDL using the <b>regsvr32</b> command from the command prompt; for example: <pre>regsvr32 "C:\Program Files\MyCompany\EAAAddIn\EAAAddIn.dll"</pre> </li> <li>If compiled as a .NET DLL, such as C# or VB.NET, register the DLL using the <b>RegAsm</b> command from the command prompt; for example:</li> </ul> |          |

| Step | Action                                                                                                                                                                                                                                                                                                                                                               | See also |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | <pre>C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\RegAsm.exe "C:\Program Files\MyCompany\EAAddin\EAAddin.dll" / codebase</pre>                                                                                                                                                                                                                                      |          |
| 3    | Place a new entry into the registry using the registry editor (run <b>regedit</b> ) so that Enterprise Architect recognizes the presence of your Add-In.                                                                                                                                                                                                             |          |
| 4    | <p>1. Add a new key value <b>EAAddins</b> under the location:</p> <ul style="list-style-type: none"> <li>• <i>HKEY_CURRENT_USER\Software\Sparx Systems</i> for single users</li> <li>• <i>HKEY_LOCAL_MACHINE\Software\Sparx Systems</i> for multiple users on a machine</li> </ul>  |          |
| 5    | <p>Add a new key under this key with the project name.</p>  <p>( <i>ProjectName</i> ) is not necessarily the name of your DLL, but the name of the Project; in Visual Basic, this is the value for the property <b>Name</b> corresponding to the project file</p>                 |          |
| 6    | <p>Specify the default value by modifying the default value of the key.</p>                                                                                                                                                                                                       |          |
| 7    | <p>Enter the value of the key by typing in the ( <i>project name</i> ) . ( <i>class name</i> ) ; for example, <i>EaRequirements.Requirements</i>, where <i>EaRequirements</i> is the project name, as shown in the example below.</p>                                                                                                                                |          |

| Step | Action                                                                             | See also |
|------|------------------------------------------------------------------------------------|----------|
|      |  |          |

### 18.3.1.1.3 Tricks and Traps

#### Topics:

| Topic                              | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Visual Basic 5/6 Users Note</b> | <p>Visual Basic 5/6 users should note that the version number of the Enterprise Architect interface is stored in the VBP project file in a form similar to the following:</p> <pre>Reference=* \ G{ 64FB2BF4- 9EFA- 11 D2- 8307- C4558600000} #2. 2#0# . . . \ . . . \ Program Files\ Sparx Systems\ EA\ EA. TLB#Enterprise Architect Object Model 2. 02</pre> <p>If you experience problems moving from one version of Enterprise Architect to another, open the VBP file in a text editor and remove this line. Then open the project in Visual Basic and use <i>Project-References</i> to create a new reference to the Enterprise Architect Object model.</p>                                                                                                                                                                                                                                                                                        |          |
| <b>Add-In Fails to Load</b>        | <p>From Enterprise Architect release 7.0, Add-Ins created before 2004 are no longer supported. If an Add-In subscribes to the <code>Addn_TmpI . t l b</code> interface (2003 style), it fails on load. In this event, contact the vendor or author of the Add-In and request an upgrade.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |          |
| <b>Holding State Information</b>   | <p>It is possible for an Add-In to hold state information, meaning that data can be stored in member variables in response to one event and retrieved in another. There are some dangers in doing this:</p> <ul style="list-style-type: none"> <li>• Enterprise Architect Automation Objects do not update themselves in response to user activity, to activity on other workstations, or even to the actions of other objects in the same automation client. Retaining handles to such objects between calls can result in the second event querying objects that have no relationship with the current state of Enterprise Architect</li> <li>• When you close Enterprise Architect, all Add-Ins are asked to shut down. If there are any external automation clients Enterprise Architect must stay active, in which case all the Add-Ins are reloaded, losing all the data</li> <li>• Enterprise Architect acting as an automation client</li> </ul> |          |

| Topic                                          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also                                                                                                                                                                                |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                | <p>does not close if an Add-In still holds a reference to it (releasing all references in the <code>Disconnect()</code> event avoids this problem)</p> <p>It is recommended that unless there is a specific reason for doing so, the Add-In should use the repository parameter and its method and properties to provide the necessary data.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                         |
| <p><b>Enterprise Architect Not Closing</b></p> | <p><b>.Net Specific Issues</b></p> <p>Automation checks the use of objects and won't enable any of them to be destroyed until they are no longer being used.</p> <p>As noted in the <b>Automation Interface</b> topic, if your automation controller was written using the .NET framework, Enterprise Architect does not close even after you release all your references to it. To force the release of the COM pointers, call the memory management functions as shown below:</p> <pre>GC. Collect(); GC. WaitForPendingFinalizers();</pre> <p>Additionally, because automation clients hook into Enterprise Architect, which creates Add-Ins which in turn hook back into Enterprise Architect, it is possible to get into a deadlock situation where Enterprise Architect and the Add-Ins won't let go of one another and keep each other active. An Add-In might retain hooks into Enterprise Architect because:</p> <ul style="list-style-type: none"> <li>• It keeps a private reference to an Enterprise Architect object (see <b>Holding State Information</b> above), or</li> <li>• It has been created by .NET and the GC mechanism hasn't got around to releasing it.</li> </ul> <p>There are two actions required to avoid deadlock situations:</p> <ul style="list-style-type: none"> <li>• Automation controllers must call <code>Repository.CloseAddIns()</code> at some point (presumably at the end of processing).</li> <li>• Add-Ins must release all references to Enterprise Architect in the <code>Disconnect()</code> event. See the <b>Add-In Events</b> topic for details.</li> </ul> <p>It is possible that your Automation client controls a running instance of Enterprise Architect where the Add-Ins have not complied with the rule above. In this case you could call <code>Repository.Exit()</code> to terminate Enterprise Architect.</p> <p><b>Miscellaneous</b></p> <p>In developing Add-Ins using the .Net framework you must select COM Interoperability in the project's properties in order for it to be recognized as an Add-In.</p> <p>Some development environments do not automatically register COM DLLs on creation. You might have to do that manually before Enterprise Architect recognizes the Add-In.</p> <p>You can use your private Add-In key (as required for Add-In deployment) to store configuration information pertinent to your Add-In.</p> | <p><a href="#">Automation Interface</a> <sup>[1826]</sup></p> <p><a href="#">Holding State Information</a> <sup>[1972]</sup></p> <p><a href="#">Add-In Events</a> <sup>[1976]</sup></p> |
| <p><b>Concurrent Calls</b></p>                 | <p>In Enterprise Architect releases up to release 7.0, there is a possibility that Enterprise Architect could call two Add-In methods concurrently if the Add-In calls:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                         |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See Also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <ul style="list-style-type: none"> <li>• A message box</li> <li>• A modal dialog</li> <li>• VB DoEvents, .NET Application DoEvents or the equivalent in other languages</li> </ul> <p>In such cases, Enterprise Architect could initiate a second Add-In method before the first returns (re-entrancy). In release 7.0. and subsequent releases, Enterprise Architect cannot make such concurrent calls</p> <p>If developing Add-Ins, ensure that the Add-In users are running Enterprise Architect release 7.0 or a later release to avoid any risk of concurrent method calls.</p> |          |

### 18.3.2 The Add-In Manager

#### Topics:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                     |        |                 |             |         |                                     |                   |         |                                     |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|--------|-----------------|-------------|---------|-------------------------------------|-------------------|---------|-------------------------------------|
| <b>General Usage</b> | <p>You can use the Add-In Manager to view what Add-Ins are available and to disable those not to be used.</p> <p>Access the <b>Manage Add-Ins</b> dialog by selecting the <b>Add-Ins   Manage Add-Ins</b> menu option.</p> <table border="1" data-bbox="478 1097 1420 1422"> <thead> <tr> <th>Available Add-Ins</th> <th>Status</th> <th>Load on Startup</th> </tr> </thead> <tbody> <tr> <td>DoDAF-MODAF</td> <td>Enabled</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Zachman Framework</td> <td>Enabled</td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table> <p>To enable an Add-In for use, select the <b>Load on Startup</b> check box. To disable an Add-In, deselect the checkbox.</p> | Available Add-Ins                   | Status | Load on Startup | DoDAF-MODAF | Enabled | <input checked="" type="checkbox"/> | Zachman Framework | Enabled | <input checked="" type="checkbox"/> |
| Available Add-Ins    | Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Load on Startup                     |        |                 |             |         |                                     |                   |         |                                     |
| DoDAF-MODAF          | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <input checked="" type="checkbox"/> |        |                 |             |         |                                     |                   |         |                                     |
| Zachman Framework    | Enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | <input checked="" type="checkbox"/> |        |                 |             |         |                                     |                   |         |                                     |

#### Notes:

- Enterprise Architect must be restarted for changes to take effect

### 18.3.3 Add-In Search

#### Topics:

| Topic                | Detail                                                                                                                                                                                           | See Also                                                                                              |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <b>General Usage</b> | Enterprise Architect enables Add-Ins to integrate with the <b>Model Search</b> . Searches can be defined that execute a method within your Add-In and display your results in an integrated way. | <a href="#">Model Search</a> <sup>[475]</sup><br><a href="#">Search Data Format</a> <sup>[1975]</sup> |

| Topic                 | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | See Also  |             |     |                                |            |                                                      |            |                                                                                                  |  |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-------------|-----|--------------------------------|------------|------------------------------------------------------|------------|--------------------------------------------------------------------------------------------------|--|
|                       | <p>The method that runs the search must be structured in the following way:</p> <p>variant &lt;method name&gt; (Rep as Repository, SearchText as String, XMLResults as String)</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Rep</td> <td>The currently open repository.</td> </tr> <tr> <td>SearchText</td> <td>An optional field that you can fill in through the M</td> </tr> <tr> <td>XMLResults</td> <td>At completion of the method, this should contain should be an XML String that conforms to the Se</td> </tr> </tbody> </table>             | Parameter | Description | Rep | The currently open repository. | SearchText | An optional field that you can fill in through the M | XMLResults | At completion of the method, this should contain should be an XML String that conforms to the Se |  |
| Parameter             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |             |     |                                |            |                                                      |            |                                                                                                  |  |
| Rep                   | The currently open repository.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |           |             |     |                                |            |                                                      |            |                                                                                                  |  |
| SearchText            | An optional field that you can fill in through the M                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |           |             |     |                                |            |                                                      |            |                                                                                                  |  |
| XMLResults            | At completion of the method, this should contain should be an XML String that conforms to the Se                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |             |     |                                |            |                                                      |            |                                                                                                  |  |
| <b>Return</b>         | The method must return a value for the results to be displayed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |           |             |     |                                |            |                                                      |            |                                                                                                  |  |
| <b>Advanced Usage</b> | <p>In addition to the displayed results, two additional hidden fields can be passed into the XML that provide special functionality.</p> <p><b>CLASSTYPE</b></p> <p>Returning a field of CLASSTYPE, containing the Object_Type value from the t_object table, displays the appropriate icon in the column you place the field.</p> <p><b>CLASSGUID</b></p> <p>Returning a field of CLASSGUID, containing an ea_guid value, enables the <b>Model Search</b> to track the object in the <b>Project Browser</b> and open the <b>Properties</b> window for the element by double-clicking in the <b>Model Search</b>.</p> |           |             |     |                                |            |                                                      |            |                                                                                                  |  |

### 18.3.3.1 XML Format (Search Data)

#### Topics:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See Also                             |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| <b>General Usage</b> | The XML below provides the format for the sSearchData parameter of the RunModelSearch method. See the <b>Repository</b> topic for more information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <a href="#">Repository</a><br>118681 |
| <b>Example Code</b>  | <pre>&lt;Report ViewData UID=\ " My Search ID \ " &gt;   &lt;!--     //The UID attribute enables XML type searches     to persist column information. That is, if you     run the search, group by column or adjust column     widths, then close the window and run the search     again, the format/organization changes are     retained. To avoid persisting column     arrangements, leave the attribute value blank or     remove it altogether.     // Use this section to declare all possible     fields - columns that appear in Enterprise     Architect's search window - that are used below     in &lt;Rows/&gt;.     // The order of the columns of information to     be appended here must match the order that the     search run in Enterprise Architect would     normally display.     // Furthermore, if you append results onto a     custom SQL Search, then the order used in your</pre> |                                      |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See Also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>Custom SQL must match the order used below.</p> <pre>--&gt; &lt;Fields&gt;   &lt;Field name="" /&gt;   &lt;Field name="" /&gt;   &lt;Field name="" /&gt;   &lt;Field name="" /&gt; &lt;/Fields&gt;  &lt;Rows&gt;   &lt;Row&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;   &lt;/Row&gt;   &lt;Row&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;   &lt;/Row&gt;   &lt;Row&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;     &lt;Field name="" value="" /&gt;   &lt;/Row&gt; &lt;/Rows&gt; &lt;/ReportViewData&gt;</pre> |          |

### 18.3.4 Add-In Events

All Enterprise Architect Add-Ins can choose to respond to the following general Add-In events:

| Topic                        | Link                                                           |
|------------------------------|----------------------------------------------------------------|
| EA_Connect                   | <a href="#">EA_Connect</a> <sup>[1977]</sup>                   |
| EA_Disconnect                | <a href="#">EA_Disconnect</a> <sup>[1977]</sup>                |
| EA_GetMenuItems              | <a href="#">EA_GetMenuItems</a> <sup>[1978]</sup>              |
| EA_MenuClick                 | <a href="#">EA_MenuClick</a> <sup>[1980]</sup>                 |
| EA_GetMenuState              | <a href="#">EA_GetMenuState</a> <sup>[1979]</sup>              |
| EA_ShowHelp                  | <a href="#">EA_ShowHelp</a> <sup>[1983]</sup>                  |
| EA_OnOutputItemClicked       | <a href="#">EA_OnOutputItemClicked</a> <sup>[1981]</sup>       |
| EA_OnOutputItemDoubleClicked | <a href="#">EA_OnOutputItemDoubleClicked</a> <sup>[1982]</sup> |

### 18.3.4.1 EA\_Connect

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also                                                                                                   |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| Details | <p>EA_Connect events enable Add-Ins to identify their type and to respond to Enterprise Architect start up.</p> <p>This event occurs when Enterprise Architect first loads your Add-In. Enterprise Architect itself is loading at this time so that while a Repository object is supplied, there is limited information that you can extract from it.</p> <p>The chief uses for EA_Connect are in initializing global Add-In data and for identifying the Add-In as an <b>MDG Add-In</b>.</p> <p>Also look at <b>EA_Disconnect</b>.</p> | <p><a href="#">MDG Add-In</a> <sup>[2042]</sup></p> <p><a href="#">EA_Disconnect</a> <sup>[1977]</sup></p> |

#### Syntax

#### Function EA\_Connect(Repository As EA.Repository) As String

The *EA\_Connect* function syntax has the following elements:

| Parameter  | Type                                            | Direction | Description                                                                                                                                                        |
|------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

String identifying a specialized type of Add-In:

| Type  | Details                                                                                  |
|-------|------------------------------------------------------------------------------------------|
| "MDG" | MDG Add-Ins receive <a href="#">MDG Events</a> <sup>[2042]</sup> and extra menu options. |
| ""    | None-specialized Add-In.                                                                 |

### 18.3.4.2 EA\_Disconnect

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                         | See Also                                            |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| Details | <p>The <i>EA_Disconnect</i> event enables the Add-In to respond to user requests to disconnect the model branch from an external project.</p> <p>This function is called when the Enterprise Architect closes. If you have stored references to Enterprise</p> | <p><a href="#">EA_Connect</a> <sup>[1977]</sup></p> |

| Topic | Detail                                                                                                                                                                                                                                                                                 | See Also |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>Architect objects (not particularly recommended anyway), you must release them here.</p> <p>In addition, .NET users must call memory management functions as shown below:</p> <pre>GC. Collect (); GC. Wait For Pending Finalizers ();</pre> <p>Also look at <b>EA_Connect</b>.</p> |          |

### Syntax

*Sub* EA\_Disconnect()

### Return Value

None.

## 18.3.4.3 EA\_GetMenuItems

### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also                                                                                                                                                               |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p>The <i>EA_GetMenuItems</i> event enables the Add-In to provide the Enterprise Architect user interface with additional Add-In menu options in various context and main menus. When a user selects an Add-In menu option, an event is raised and passed back to the Add-In that originally defined that menu option.</p> <p>This event is raised just before Enterprise Architect has to show particular menu options to the user, and its use is described in the <b>Define Menu Items</b> topic.</p> | <p><a href="#">Define Menu Items</a><sup>[1969]</sup></p> <p><a href="#">EA_MenuClick</a><sup>[1980]</sup></p> <p><a href="#">EA_GetMenuState</a><sup>[1979]</sup></p> |

### Syntax

*Function* EA\_GetMenuItems(*Repository As EA.Repository, MenuLocation As String, MenuName As String*) *As Variant*

The *EA\_GetMenuItems* function syntax has the following elements:

| Parameter    | Type                                            | Direction | Description                                                                                                                                                        |
|--------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MenuLocation | String                                          |           | String representing the part of the user interface that brought up the menu. Can be <i>TreeView</i> , <i>MainMenu</i> or <i>Diagram</i> .                          |
| MenuName     | String                                          |           | The name of the parent menu for which sub-items are to be defined. In the case of the top-level menu it is an empty string.                                        |
| Repository   | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

One of the following types:

- A string indicating the label for a single menu option.
- An array of strings indicating a multiple menu options.
- Empty (Visual Basic/VB.NET) or null (C#) to indicate that no menu should be displayed

In the case of the top-level menu it should be a single string or an array containing only one item, or Empty/null.

### 18.3.4.4 EA\_GetMenuState

#### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | See Also                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p>The <i>EA_GetMenuState</i> event enables the Add-In to set a particular menu option to either enabled or disabled. This is useful when dealing with locked packages and other situations where it is convenient to show a menu option, but not enable it for use.</p> <p>This event is raised just before Enterprise Architect has to show particular menu options to the user. Its use is described in the <b>Define Menu Items</b> topic.</p> <p>Also look at <b>EA_GetMenuItems</b>.</p> | <p><a href="#">Define Menu Items</a><br/>[1969]</p> <p><a href="#">EA_GetMenuItems</a><br/>[1978]</p> |

#### Syntax

*Sub EA\_GetMenuState(Repository as EA.Repository, MenuLocation As String, MenuName as String, ItemName as String, IsEnabled as Boolean, IsChecked as Boolean)*

The *EA\_GetMenuState* function syntax has the following elements:

| Parameter           | Type                                    | Direction | Description                                                                                                                                                               |
|---------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>IsChecked</b>    | <i>Boolean</i>                          |           | Boolean. Set to <b>True</b> to check this particular menu option.                                                                                                         |
| <b>IsEnabled</b>    | <i>Boolean</i>                          |           | Boolean. Set to <b>False</b> to disable this particular menu option.                                                                                                      |
| <b>ItemName</b>     | <i>String</i>                           |           | The name of the option actually clicked, for example, <i>Create a New Invoice</i> .                                                                                       |
| <b>MenuLocation</b> | <i>String</i>                           |           | String representing the part of the user interface that brought up the menu. Can be <i>TreeView</i> , <i>MainMenu</i> or <i>Diagram</i> .                                 |
| <b>MenuName</b>     | <i>String</i>                           |           | The name of the parent menu for which sub-items must be defined. In the case of the top-level menu it is an empty string.                                                 |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br>[1858] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

None.

### 18.3.4.5 EA\_MenuClick

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also                                                                                                   |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| Details | <p><i>EA_MenuClick</i> events are received by an Add-In in response to user selection of a menu option.</p> <p>The event is raised when the user clicks on a particular menu option. When a user clicks on one of your non-parent menu options, your Add-In receives a <i>MenuClick</i> event, defined as follows:</p> <pre>Sub EA_MenuClick(Repository As EA.Repository, ByVal MenuName As String, ByVal ItemName As String)</pre> <p>The code below illustrates an example of use:</p> <pre>    If MenuName = "- &amp;Diagram" And     ItemName = "&amp;Properties" then         MsgBox Repository.     Get Current Diagram Name, vbInformation     Else         MsgBox "Not Implemented",         vbCritical     End If</pre> <p>Notice that your code can directly access Enterprise Architect data and UI elements using <b>Repository</b> methods.</p> <p>Also look at <b>EA_GetMenuItems</b>.</p> | <p><a href="#">Repository</a><sup>[1855]</sup></p> <p><a href="#">EA_GetMenuItems</a><sup>[1978]</sup></p> |

#### Syntax

**Sub EA\_MenuClick(Repository As EA.Repository, MenuLocation As String, MenuName As String, ItemName As String)**

The *EA\_GetMenuClick* function syntax has the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                               |
|-------------------|-------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ItemName</b>   | <i>String</i>                                   |           | The name of the option actually clicked, for example, <i>Create a New Invoice</i> .                                                                                       |
| <b>MenuName</b>   | <i>String</i>                                   |           | The name of the parent menu for which sub-items are to be defined. In the case of the top-level menu it is an empty string.                                               |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

None.

### 18.3.4.6 EA\_OnOutputItemClicked

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See Also                                                                                  |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Details | <p><i>EA_OnOutputItemClicked</i> events inform Add-Ins that the user has clicked on a list entry in the system tab or one of the user defined output tabs.</p> <p>Usually an Add-In responds to this event in order to capture activity on an output tab they had previously created through a call to <i>Repository.AddTab()</i>.</p> <p>Note that every loaded Add-In receives this event for every click on an output tab in Enterprise Architect - irrespective of whether the Add-In created that tab. Add-Ins should therefore check the <b>TabName</b> parameter supplied by this event to ensure that they are not responding to other Add-Ins' events.</p> <p>Also look at <b>EA_OnOutputItemDoubleClicked</b>.</p> | <a href="#">EA_OnOutputItemDoubleClicked</a><br><a href="#">Clicked</a> <sup>[1982]</sup> |

#### Syntax

**EA\_OnOutputItemClicked**(*Repository As EA.Repository, TabName As String, LineText As String, ID As Long*)

The *EA\_OnOutputItemClicked* function syntax has the following elements:

| Parameter         | Type                                               | Direction | Description                                                                                                                                                               |
|-------------------|----------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ID</b>         | <i>Long</i>                                        | IN        | The ID value specified in the original call to <i>Repository.WriteOutput()</i> .                                                                                          |
| <b>LineText</b>   | <i>String</i>                                      | IN        | The text that had been supplied as the String parameter in the original call to <i>Repository.WriteOutput()</i> .                                                         |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| <b>TabName</b>    | <i>String</i>                                      | IN        | The name of the tab that the click occurred in. Usually this would have been created through <i>Repository.AddTab()</i> .                                                 |

#### Return Value

None.

### 18.3.4.7 EA\_OnOutputItemDoubleClicked

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also                                                        |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| Details | <p><i>EA_OnOutputItemDoubleClicked</i> events informs Add-Ins that the user has used the mouse to double-click on a list entry in one of the user-defined output tabs.</p> <p>Usually an Add-In responds to this event in order to capture activity on an output tab they had previously created through a call to <i>Repository.AddTab()</i>.</p> <p>Note that every loaded Add-In receives this event for every double-click on an output tab in Enterprise Architect - irrespective of whether the Add-In created that tab. Add-Ins should therefore check the <b>TabName</b> parameter supplied by this event to ensure that they are not responding to other Add-Ins' events.</p> <p>Also look at <b>EA_OnOutputItemClicked</b>.</p> | <a href="#">EA_OnOutputItemClicked</a><br><small>[1987]</small> |

#### Syntax

**EA\_OnOutputItemDoubleClicked**(*Repository As EA.Repository, TabName As String, LineText As String, ID As Long*)

The *EA\_OnOutputItemClicked* function syntax contains the following elements:

| Parameter         | Type                                                   | Direction | Description                                                                                                                                                               |
|-------------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ID</b>         | <i>Long</i>                                            | IN        | The ID value specified in the original call to <i>Repository.WriteOutput()</i> .                                                                                          |
| <b>LineText</b>   | <i>String</i>                                          | IN        | The text that had been supplied as the String parameter in the original call to <i>Repository.WriteOutput()</i> .                                                         |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| <b>TabName</b>    | <i>String</i>                                          | IN        | The name of the tab that the click occurred in. Usually this would have been created through <i>Repository.AddTab()</i> .                                                 |

#### Return Value

None.

### 18.3.4.8 EA\_ShowHelp

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also                                                              |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| Details | <p>The <i>EA_ShowHelp</i> event enables the Add-In to show a help topic for a particular menu option. When the user has an Add-In menu option selected, pressing ( <b>F1</b> ) can be delegated to the required Help topic by the Add-In and a suitable help message shown.</p> <p>This event is raised when the user presses ( <b>F1</b> ) on a menu option that is not a parent menu.</p> <p>Also look at <i>EA_GetMenuItems</i>.</p> | <a href="#">EA_GetMenuItem</a><br><a href="#">s</a> <sup>[1978]</sup> |

#### Syntax

*Sub EA\_ShowHelp(Repository as EA.Repository, MenuLocation As String, MenuName as String, ItemName as String)*

The *EA\_ShowHelp* function syntax contains the following elements:

| Parameter    | Type                                               | Direction | Description                                                                                                                                                               |
|--------------|----------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ItemName     | String                                             |           | The name of the option actually clicked, for example, <b>Create a New Invoice</b> .                                                                                       |
| MenuLocation | String                                             |           | String representing the part of the user interface that brought up the menu. Can be <b>Treeview</b> , <b>MainMenu</b> or <b>Diagram</b> .                                 |
| MenuName     | String                                             |           | The name of the parent menu for which sub-items are to be defined. In the case of the top-level menu it is an empty string.                                               |
| Repository   | <a href="#">EA.Repository</a><br><sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

None.

### 18.3.5 Broadcast Events

#### Overview

The following general Broadcast events are sent to all loaded Add-Ins. For an Add-In to receive the event, they must first implement the required automation event interface. If Enterprise Architect detects that the Add-In has the required interface, the event is dispatched to the Add-In.

| Topic                            | Link                                                               |
|----------------------------------|--------------------------------------------------------------------|
| Add-In Licence Management Events | <a href="#">Add-In Licence Management Events</a> <sup>[1984]</sup> |

| Topic                                  | Link                                                            |
|----------------------------------------|-----------------------------------------------------------------|
| Compartment Events                     | <a href="#">Compartment Events</a> <sup>[1986]</sup>            |
| Context Item Events                    | <a href="#">Context Item Events</a> <sup>[1989]</sup>           |
| File Close Event                       | <a href="#">File Close Event</a> <sup>[1993]</sup>              |
| File New Event                         | <a href="#">File New Event</a> <sup>[1993]</sup>                |
| File Open Event                        | <a href="#">File Open Event</a> <sup>[1994]</sup>               |
| Model Validation Broadcasts            | <a href="#">Model Validation Broadcasts</a> <sup>[1999]</sup>   |
| Post Close Diagram Event               | <a href="#">Post Close Diagram Event</a> <sup>[1995]</sup>      |
| Post Initialization Event              | <a href="#">EA_OnPostInitalized</a> <sup>[1995]</sup>           |
| Post New Events                        | <a href="#">Post-New Events</a> <sup>[2010]</sup>               |
| Post Open Diagram Event                | <a href="#">Post Open Diagram Event</a> <sup>[1996]</sup>       |
| Pre-Deletion Events                    | <a href="#">Pre-Deletion Events</a> <sup>[2017]</sup>           |
| Pre-Exit Instance (not currently used) | <a href="#">PreExit Instance</a> <sup>[1997]</sup>              |
| Pre-New Events                         | <a href="#">Pre-New Events</a> <sup>[2023]</sup>                |
| Retrieve Model Template Event          | <a href="#">Retrieve Model Template Event</a> <sup>[1998]</sup> |
| Tagged Value Broadcasts                | <a href="#">Tagged Value Broadcasts</a> <sup>[2031]</sup>       |
| Technology Events                      | <a href="#">Technology Events</a> <sup>[2034]</sup>             |
| Transformation Event                   | <a href="#">Transformation Event</a> <sup>[1997]</sup>          |

**MDG Events** add quite a number of additional events, but the Add-In must first have registered as an MDG-style Add-In, rather than as a generic Add-In.

**Learn More:**

- [MDG Events](#)<sup>[2042]</sup>

### 18.3.5.1 Add-In License Management Events

Enterprise Architect Add-ins can respond to the following events associated with Add-in License Management:

| Topic                         | Link                                                            |
|-------------------------------|-----------------------------------------------------------------|
| EA_AddinLicenseValidate       | <a href="#">EA_AddinLicenseValidate</a> <sup>[1985]</sup>       |
| EA_AddinLicenseGetDescription | <a href="#">EA_AddinLicenseGetDescription</a> <sup>[1986]</sup> |

### 18.3.5.1.1 EA\_AddinLicenseValidate

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See Also                                                        |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| Details | <p><b>EA_AddinLicenseValidate</b> is broadcast to all Add-Ins, providing them with a chance to use the Add-In key to determine the level of functionality to provide. The event is broadcast in response to a non-standard Enterprise Architect key being entered into the <b>License Management</b> dialog.</p> <p>For the Add-In to validate itself against this key, the Add-In's <b>EA_AddinLicenseValidate</b> handler should return <b>true</b> to advise Enterprise Architect that the license has been validated. The <b>EA_AddinLicenseValidate</b> event is broadcast to all Add-Ins, therefore one license can validate many Add-Ins.</p> <p>If an Add-In elects to handle a license key by returning true to <b>EA_AddinLicenseValidate</b>, it is called upon to provide a description of the license key through the <b>EA_AddinLicenseGetDescription</b> event. If more than one Add-In elects to handle a license key, the first Add-In that returns <b>true</b> to <b>EA_AddinLicenseValidate</b> is queried for the license key description.</p> | <a href="#">EA_AddinLicenseGetDescription</a> <sup>[1986]</sup> |

#### Syntax

**Function** EA\_AddinLicenseValidate(*Repository As EA.Repository, AddinKey As String*) As Boolean

The *EA\_AddinLicenseValidate* function syntax contains the following elements.

| Parameter  | Type                                            | Direction | Description                                                                                                                                                        |
|------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AddinKey   | String                                          | IN        | The Add-in license key that has been entered in Enterprise Architect's <b>License Management</b> dialog.                                                           |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

For the Add-in to validate against this key it should return **true** to indicate to Enterprise Architect that the key is valid and has been handled.

### 18.3.5.1.2 EA\_AddinLicenseGetDescription

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                  | See Also |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | Before Enterprise Architect's <b>License Management</b> dialog is displayed, <b>EA_AddinLicenseGetDescription</b> is sent once for each Add-In key to the first Add-In that elected to handle that key. The value returned by <b>EA_AddinLicenseGetDescription</b> is used by Enterprise Architect as the key's plain text description. |          |

#### Syntax

**Function** EA\_AddinLicenseGetDescription (*Repository as EA.Repository, AddinKey as String*) As String

The *EA\_OnPostInitialized* function syntax contains the following elements.

| Parameter  | Type                                            | Direction | Description                                                                                                                                                        |
|------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AddinKey   | String                                          | IN        | The Add-In license key that Enterprise Architect requires a description for.                                                                                       |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

A String containing a plain text description of the provided AddinKey.

### 18.3.5.2 Compartment Events

Enterprise Architect Add-Ins can respond to the following events associated with user-generated element compartments:

| Topic                         | Link                                                            |
|-------------------------------|-----------------------------------------------------------------|
| EA_QueryAvailableCompartments | <a href="#">EA_QueryAvailableCompartments</a> <sup>[1986]</sup> |
| EA_GetCompartmentData         | <a href="#">EA_GetCompartmentData</a> <sup>[1987]</sup>         |

#### 18.3.5.2.1 EA\_QueryAvailableCompartments

#### Topics:

| Topic   | Detail                                                     | See Also                             |
|---------|------------------------------------------------------------|--------------------------------------|
| Details | This event occurs when Enterprise Architect's diagrams are | <a href="#">EA_GetCompartmentDat</a> |

| Topic | Detail                                                                                                                                                                                                              | See Also                            |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
|       | refreshed. It is a request for the Add-In to provide a list of user-defined compartments. The <b>EA_GetCompartmentData</b> event then queries each object for the data to display in each user-defined compartment. | <a href="#">a</a> <sup>[1987]</sup> |

### Syntax

#### Function EA\_QueryAvailableCompartments(*Repository As EA.Repository*) As Variant

The *EA\_QueryAvailableCompartments* function syntax contains the following elements:

| Parameter  | Type                                            | Direction | Description                                                                                                                                                               |
|------------|-------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

A *String* containing a comma-separated list of user-defined compartments.

### Example

```
Function EA_QueryAvailableCompartments(Repository As EA.Repository) As Variant
 Dim sReturn As String
 sReturn = ""
 If m_FirstCompartmentVisible = True Then
 sReturn = sReturn + "first,"
 End If
 If m_SecondCompartmentVisible = True Then
 sReturn = sReturn + "second,"
 End If
 If m_ThirdCompartmentVisible = True Then
 sReturn = sReturn + "third,"
 End If

 If Len(sReturn) > 0 Then
 sReturn = Left(sReturn, Len(sReturn) - 1)
 End If

 EA_QueryAvailableCompartments = sReturn
End Function
```

#### 18.3.5.2.2 EA\_GetCompartmentData

### Topics:

| Topic   | Detail                                                                                                                                                 | See Also |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | This event occurs when Enterprise Architect is instructed to redraw an element. It requests that the Add-In provide the data to populate the element's |          |

| Topic | Detail       | See Also |
|-------|--------------|----------|
|       | compartment. |          |

### Syntax

**Function** EA\_GetCompartmentData(*Repository* As EA.Repository, *sCompartment* As String, *sGUID* As String, *oType* As EA.ObjectType) As Variant

The *EA\_QueryAvailableCompartments* function syntax contains the following elements:

| Parameter           | Type                                    | Direction | Description                                                                                                                                                               |
|---------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>oType</b>        | <i>ObjectType</i>                       | IN        | The type of the element for which data is being requested.                                                                                                                |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| <b>sCompartment</b> | <i>String</i>                           | IN        | The name of the compartment for which data is being requested.                                                                                                            |
| <b>sGUID</b>        | <i>String</i>                           | IN        | The GUID of the element for which data is being requested.                                                                                                                |

### Return Value

Variant containing a formatted string. See the example below to understand the format.

### Example

```
Function EA_GetCompartmentData(Repository As EA.Repository, sCompartment As String, sGUID As String, oType As EA.ObjectType) As Variant
```

```
 If Repository Is Nothing Then
 Exit Function
 End If
```

```
 Dim sCompartmentData As String
 Dim oXML As MSXML2.DOMDocument
 Dim Nodes As MSXML2.IXMLDOMNodeList
 Dim Node1 As MSXML2.IXMLDOMNode
 Dim Node As MSXML2.IXMLDOMNode
 Dim sData As String
```

```
 sCompartmentData = ""
 Set oXML = New MSXML2.DOMDocument
 sData = ""
```

```
 On Error GoTo ERR_GetCompartmentData
```

```
 oXML.LoadXML (Repository.GetTreeXMLByGUID(sGUID))
 Set Node1 = oXML.selectSingleNode("//ModelItem")
```

```
 If Node1 Is Nothing Then
 Exit Function
 End If
```

```
 sCompartmentData = sCompartmentData + " Name=" + sCompartment + ";"
```

```

sCompartmentData = sCompartmentData + " Owner GUID=" + sGUID + "; "
sCompartmentData = sCompartmentData +
" Options=SkipIfOnDiagram&_eq_^1&_sc_^"

Select Case sCompartment
Case "parts"
 Set Nodes = Node1.selectNodes(" ModelItem(@Metatype="" Part "") ")
 For Each Node In Nodes
 sData = sData + "Data&_eq_^" + Node.Attributes.getNamedItem
(" Name").nodeValue + "&_sc_^"
 sData = sData + "GUID&_eq_^" + Node.Attributes.getNamedItem
(" GUID").nodeValue + "&_sc_^,"
 Next

Case "ports"
 Set Nodes = Node1.selectNodes(" ModelItem(@Metatype="" Port "") ")
 For Each Node In Nodes
 sData = sData + "Data&_eq_^" + Node.Attributes.getNamedItem
(" Name").nodeValue + "&_sc_^"
 sData = sData + "GUID&_eq_^" + Node.Attributes.getNamedItem
(" GUID").nodeValue + "&_sc_^,"
 Next

End Select

' If there's no data to display, then don't return any compartment
data
If sData <> "" Then
 sCompartmentData = sCompartmentData + " Compartment Data=" + sData +
"; "
Else
 sCompartmentData = ""
End If

EA_GetCompartmentData = sCompartmentData
Exit Function

ERR_GetCompartmentData:
EA_GetCompartmentData = ""

End Function

```

### 18.3.5.3 Context Item Events

Enterprise Architect Add-Ins can respond to the following events associated with changing context:

| Topic                          | Link                                                                       |
|--------------------------------|----------------------------------------------------------------------------|
| EA_OnContextItemChanged        | <a href="#">EA_OnContextItemChanged</a><br>[1990]                          |
| EA_OnContextItemDoubleClicked  | <a href="#">EA_OnContextItemDoubleClic</a><br><a href="#">ked</a> [1991]   |
| EA_OnNotifyContextItemModified | <a href="#">EA_OnNotifyContextItem Modifi</a><br><a href="#">ed</a> [1992] |

### 18.3.5.3.1 EA\_OnContextItemChanged

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also                                                                                                                                                                                            |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><b>EA_OnContextItemChanged</b> notifies Add-Ins that a different item is now in context.</p> <p>This event occurs after a user has selected an item anywhere in the Enterprise Architect GUI. Add-Ins that require knowledge of the current item in context can subscribe to this broadcast function. If <b>ot = otRepository</b>, then this function behaves the same as <b>EA_FileOpen</b>.</p> <p>Also look at <b>EA_OnContextItemDoubleClicked</b> and <b>EA_OnNotifyContextItemModified</b>.</p> | <p><a href="#">EA_FileOpen</a> <sup>[1994]</sup></p> <p><a href="#">EA_OnContextItemDoubleClicked</a> <sup>[1997]</sup></p> <p><a href="#">EA_OnNotifyContextItemModified</a> <sup>[1992]</sup></p> |

#### Syntax

**Sub EA\_OnContextItemChanged(Repository As EA.Repository, GUID As String, ot as EA.ObjectType)**

The *EA\_OnContextItemChanged* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GUID</b>       | <i>String</i>                                   | IN        | <p>Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:</p> <p><i>ot (ObjectType)</i> - GUID value</p> <p><i>otElement</i> - <i>Element.ElementGUID</i></p> <p><i>otPackage</i> - <i>Package.PackageGUID</i></p> <p><i>otDiagram</i> - <i>Diagram.DiagramGUID</i></p> <p><i>otAttribute</i> - <i>Attribute.AttributeGUID</i></p> <p><i>otMethod</i> - <i>Method.MethodGUID</i></p> <p><i>otConnector</i> - <i>Connector.ConnectorGUID</i></p> <p><i>otRepository</i> - NOT APPLICABLE, GUID is an empty string</p> |
| <b>ot</b>         | <i>EA.ObjectType</i>                            | IN        | Specifies the type of the new context item.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                                                                                                                                       |

#### Return Value

None.

### 18.3.5.3.2 EA\_OnContextItemDoubleClicked

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See Also                                                                                                                               |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><b>EA_OnContextItemDoubleClicked</b> notifies Add-Ins that the user has double-clicked the item currently in context.</p> <p>This event occurs when a user has double-clicked (or pressed ( <b>Enter</b> ) ) on the item in context, either in a diagram or in the <b>Project Browser</b>. Add-Ins to handle events can subscribe to this broadcast function.</p> <p>Also look at <b>EA_OnContextItemChanged</b> and <b>EA_OnNotifyContextItemModified</b></p> | <p><a href="#">EA_OnContextItemChanged</a><sup>[1990]</sup></p> <p><a href="#">EA_OnNotifyContextItemModified</a><sup>[1992]</sup></p> |

#### Syntax

Function **EA\_OnContextItemDoubleClicked**(Repository As EA.Repository, GUID As String, ot as EA.ObjectType)

The *EA\_OnContextItemDoubleClicked* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GUID</b>       | String                                          | IN        | <p>Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:</p> <p><i>ot</i> (ObjectType) - GUID value</p> <p><i>otElement</i> - Element.ElementGUID</p> <p><i>otPackage</i> - Package.PackageGUID</p> <p><i>otDiagram</i> - Diagram.DiagramGUID</p> <p><i>otAttribute</i> - Attribute.AttributeGUID</p> <p><i>otMethod</i> - Method.MethodGUID</p> <p><i>otConnector</i> - Connector.ConnectorGUID</p> |
| <b>ot</b>         | EA.ObjectType                                   | IN        | Specifies the type of the new context item.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                               |

#### Return Value

Return **True** to notify Enterprise Architect that the double-click event has been handled by an Add-In. Return **False** to enable Enterprise Architect to continue processing the event.

### 18.3.5.3.3 EA\_OnNotifyContextItemModified

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                     | See Also                                                                                                                              |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><b>EA_OnNotifyContextItemModified</b> notifies Add-Ins that the current context item has been modified.</p> <p>This event occurs when a user has modified the context item. Add-Ins that require knowledge of when an item has been modified can subscribe to this broadcast function.</p> <p>Also look at <b>EA_OnContextItemChanged</b> and <b>EA_OnContextItemDoubleClicked</b>.</p> | <p><a href="#">EA_OnContextItemChanged</a><sup>[1990]</sup></p> <p><a href="#">EA_OnContextItemDoubleClicked</a><sup>[1991]</sup></p> |

#### Syntax

**Sub EA\_OnNotifyContextItemModified(Repository As EA.Repository, GUID As String, ot as EA.ObjectType)**

The *EA\_OnNotifyContextItemModified* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|-------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GUID</b>       | <i>String</i>                                   | IN        | <p>Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:</p> <ul style="list-style-type: none"> <li>• <i>ot(ObjectType)</i> - GUID value</li> <li>• <i>otElement</i> - <i>Element.ElementGUID</i></li> <li>• <i>otPackage</i> - <i>Package.PackageGUID</i></li> <li>• <i>otDiagram</i> - <i>Diagram.DiagramGUID</i></li> <li>• <i>otAttribute</i> - <i>Attribute.AttributeGUID</i></li> <li>• <i>otMethod</i> - <i>Method.MethodGUID</i></li> <li>• <i>otConnector</i> - <i>Connector.ConnectorGUID</i></li> </ul> |
| <b>ot</b>         | <i>EA.ObjectType</i>                            | IN        | Specifies the type of the new context item.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                                                                                                                                      |

#### Return Value

None.

### 18.3.5.4 EA\_FileClose

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also                                                                          |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| Details | <p>The EA_FileClose event enables the Add-In to respond to a File Close event. When Enterprise Architect closes an opened Model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>This event occurs when the model currently opened within Enterprise Architect is about to be closed (when another model is about to be opened or when Enterprise Architect is about to shutdown).</p> <p>Also look at <b>EA_FileOpen</b> and <b>EA_FileNew</b>.</p> | <a href="#">EA_FileOpen</a><br>[1994]<br><br><a href="#">EA_FileNew</a><br>[1993] |

#### Syntax

##### Sub EA\_FileClose(Repository As EA.Repository)

The EA\_FileClose function syntax contains the following elements:

| Parameter  | Type                                    | Direction | Description                                                                                                                                                            |
|------------|-----------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a><br>[1855] | IN        | An EA.Repository object representing the Enterprise Architect model about to be closed. Poll its members to retrieve model data and user interface status information. |

#### Return Value

None.

### 18.3.5.5 EA\_FileNew

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                       | See Also                                                                            |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Details | <p>The EA_FileNew event enables the Add-In to respond to a File New event. When Enterprise Architect creates a new model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>The event occurs when the model being viewed by the Enterprise Architect user changes, for whatever reason (through user interaction or Add-In activity).</p> | <a href="#">EA_FileClose</a><br>[1993]<br><br><a href="#">EA_FileOpen</a><br>[1994] |

#### Syntax

##### Sub EA\_FileNew(Repository As EA.Repository)

The *EA\_FileNew* function syntax contains the following elements:

| Parameter  | Type                                    | Direction | Description                                                                                                                                                               |
|------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

None.

### 18.3.5.6 *EA\_FileOpen*

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also                                                                                    |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| Details | <p>The <i>EA_FileOpen</i> event enables the Add-In to respond to a <i>File Open</i> event. When Enterprise Architect opens a new model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>The event occurs when the model being viewed by the Enterprise Architect user changes, for whatever reason (through user interaction or Add-In activity).</p> <p>Also look at <b>EA_FileClose</b> and <b>EA_FileNew</b>.</p> | <p><a href="#">EA_FileClose</a><br/>[1993]</p> <p><a href="#">EA_FileNew</a><br/>[1993]</p> |

#### Syntax

##### *Sub EA\_FileOpen(Repository As EA.Repository)*

The *EA\_FileOpen* function syntax contains the following elements:

| Parameter  | Type                                    | Direction | Description                                                                                                                                                               |
|------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

None.

### 18.3.5.7 EA\_OnPostCloseDiagram

#### Topics:

| Topic   | Detail                                                                                                                      | See Also                                               |
|---------|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| Details | <i>EA_OnPostCloseDiagram</i> notifies Add-Ins that a diagram has been closed.<br>Also look at <i>EA_OnPostOpenDiagram</i> . | <a href="#">EA_OnPostOpenDiagram</a> <sup>[1996]</sup> |

#### Syntax

Function *EA\_OnPostCloseDiagram(Repository As EA.Repository, DiagramID As Integer)*

The *EA\_OnPostCloseDiagram* function syntax contains the following elements:

| Parameter  | Type                                            | Direction | Description                                                                                                                                                                   |
|------------|-------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DiagramID  | <i>Integer</i>                                  | IN        | Contains the Diagram ID of the diagram that was closed.                                                                                                                       |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the Enterprise Architect model about to be closed. Poll its members to retrieve model data and user interface status information. |

#### Return Value

None.

### 18.3.5.8 EA\_OnPostInitialized

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                             | See Also                                                     |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| Details | <i>EA_OnPostInitialized</i> notifies Add-Ins that the Repository object has finished loading and any necessary initialization steps can now be performed on the object<br>For example, the Add-In can create an Output tab using <i>Repository.CreateOutputTab</i> | <a href="#">Repository.CreateOutputTab</a> <sup>[1860]</sup> |

#### Syntax

Sub *EA\_OnPostInitialized(Repository As EA.Repository)*

The *EA\_OnPostInitialized* function syntax contains the following elements:

| Parameter  | Type                                            | Direction | Description                                                    |
|------------|-------------------------------------------------|-----------|----------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open |

| Parameter | Type | Direction | Description                                                                                                 |
|-----------|------|-----------|-------------------------------------------------------------------------------------------------------------|
|           |      |           | Enterprise Architect model<br>Poll its members to retrieve model data and user interface status information |

#### Return Value

None

### 18.3.5.9 EA\_OnPostOpenDiagram

#### Topics:

| Topic   | Detail                                                                                                                      | See Also                                                |
|---------|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| Details | <b>EA_OnPostOpenDiagram</b> notifies Add-Ins that a diagram has been opened.<br>Also look at <b>EA_OnPostCloseDiagram</b> . | <a href="#">EA_OnPostCloseDiagram</a> <sup>[1995]</sup> |

#### Syntax

**Function** EA\_OnPostOpenDiagram(*Repository As EA.Repository, DiagramID As Integer*)

The *EA\_OnPostOpenDiagram* function syntax contains the following elements:

| Parameter  | Type                                            | Direction | Description                                                                                                                                                               |
|------------|-------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DiagramID  | <i>Integer</i>                                  | IN        | Contains the Diagram ID of the diagram that was opened.                                                                                                                   |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

None.

### 18.3.5.10 EA\_OnPostTransform

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                          | See Also |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnPostTransform</b> notifies Add-Ins that an MDG transformation has taken place with the output in the specified target package</p> <p>This event occurs when a user runs an MDG transform on one or more target packages; the notification is provided for each transform/target package immediately after all transform processes have completed</p> |          |

#### Syntax

**Function** EA\_OnPostTransform(*Repository As EA.Repository, Info As EA.EventProperties*) *As Boolean*

The *EA\_OnPostTransform* function syntax contains the following elements:

| Parameter  | Type                                                        | Direction | Description                                                                                                                                                                                                                                                                                                                             |
|------------|-------------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | <p>Contains the following <i>EventProperty Objects</i> for the transform performed:</p> <ul style="list-style-type: none"> <li><i>Transform</i>: A string value corresponding to the name of the transform used</li> <li><i>PackageID</i>: A long value corresponding to <i>Package.PackageID</i> of the destination package</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | <p>An EA.Repository object representing the currently open Enterprise Architect model</p> <p>Poll its members to retrieve model data and user interface status information</p>                                                                                                                                                          |

#### Return Value

Reserved for future use.

### 18.3.5.11 EA\_OnPreExitInstance

#### Details:

*EA\_OnPreExitInstance* is not currently used.

#### Syntax:

**Sub** EA\_OnPreExitInstance(*Repository As EA.Repository*)

The *EA\_OnPreExitInstance* function syntax contains the following element:

| Parameter  | Type                                                   | Direction | Description                                                                                                                                                               |
|------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

**Return Value:**

None.

**18.3.5.12 EA\_OnRetrieveModelTemplate****Topics:**

| Topic   | Detail                                                                                                                                                                                                                                                          | See Also                                                             |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| Details | <b>EA_OnRetrieveModelTemplate</b> requests that an Add-In pass a model template to Enterprise Architect<br><br>This event occurs when a user executes the <b>Add a New Model Using Wizard</b> command to add a model that has been defined by an MDG Technology | <a href="#">Incorporate Model Templates</a><br><small>[1087]</small> |

**Syntax**

**Function** EA\_OnRetrieveModelTemplate(*Repository As EA.Repository, sLocation As String*) **As String**

The *EA\_OnRetrieveModelTemplate* function syntax contains the following elements:

| Parameter  | Type                                                   | Direction | Description                                                                                                                                                                    |
|------------|--------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model<br><br>Poll its members to retrieve model data and user interface status information |
| sLocation  | <i>String</i>                                          | IN        | The name of the template requested; this should match the <i>location</i> attribute in the <i>&lt;ModelTemplates&gt;</i> section of an MDG Technology File                     |

**Return Value**

Return a string containing the XML export of the model that is being used as a template

**Example**

```
Public Function EA_OnRetrieveModelTemplate(ByRef Rep As EA.Repository,
ByRef sLocation As String) As String
 Dim sTemplate As String
 Select Case sLocation
 Case "Templates\ Template1.xml "
 sTemplate = My.Resources.Template1
```

```

Case " Templates\ Template2.xml "
 sTemplate = My.Resources.Template2
Case " Templates\ Template3.xml "
 sTemplate = My.Resources.Template3
Case Else
 MsgBox(" Path for " & sLocation & " not found")
 sTemplate = ""
End Select
EA_OnRetrieveModelTemplate = sTemplate
End Function

```

### 18.3.5.13 Model Validation Broadcasts

#### Perform Model Validation from an Add-In

Using Enterprise Architect broadcasts, it is possible to define a set of rules that are evaluated when the user instructs Enterprise Architect to perform model validation. An Add-In that performs model validation would involve the following broadcast events:

| Topic                    | Detail                                                                                                                              | Link                                                       |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| EA_OnInitializeUserRules | EA_OnInitializeUserRules is intercepted in order to define rule categories and rules                                                | <a href="#">EA_OnInitializeUserRules</a> <sup>[2001]</sup> |
| EA_OnStartValidation     | EA_OnStartValidation can be intercepted to perform any required processing prior to validation                                      | <a href="#">EA_OnStartValidation</a> <sup>[2001]</sup>     |
| Validate Request         | The following functions intercept each request to validate an individual element, package, diagram, connector, attribute and method |                                                            |
|                          | EA_OnRunElementRule                                                                                                                 | <a href="#">EA_OnRunElementRule</a> <sup>[2001]</sup>      |
|                          | EA_OnRunPackageRule                                                                                                                 | <a href="#">EA_OnRunPackageRule</a> <sup>[2002]</sup>      |
|                          | EA_OnRunDiagramRule                                                                                                                 | <a href="#">EA_OnRunDiagramRule</a> <sup>[2002]</sup>      |
|                          | EA_OnRunConnectorRule                                                                                                               | <a href="#">EA_OnRunConnectorRule</a> <sup>[2003]</sup>    |
|                          | EA_OnRunAttributeRule                                                                                                               | <a href="#">EA_OnRunAttributeRule</a> <sup>[2004]</sup>    |
|                          | EA_OnRunMethodRule                                                                                                                  | <a href="#">EA_OnRunMethodRule</a> <sup>[2005]</sup>       |
| EA_OnEndValidation       | can be intercepted to perform any required clean-up after validation has completed                                                  | <a href="#">EA_OnEndValidation</a> <sup>[2001]</sup>       |

#### Learn More:

- Also consider the [Model Validation Example](#) <sup>[2006]</sup>

### 18.3.5.13.1 EA\_OnInitializeUserRules

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also                                                         |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| Details | <p><b>EA_OnInitializeUserRules</b> is called on Enterprise Architect start-up and requests that the Add-In provide Enterprise Architect with a rule category and list of rule IDs for model validation.</p> <p>This function must be implemented by any Add-In that is to perform its own model validation. It must call <i>Project.DefineRuleCategory</i> once and <i>Project.DefineRule</i> for each rule; these functions are described in the <b>Project Interface</b> section.</p> | <a href="#">Project Interface Class</a><br><small>[1929]</small> |

#### Syntax

##### Sub EA\_OnInitializeUserRules(Repository As EA.Repository)

The *EA\_OnInitializeUserRules* function syntax contains the following elements:

| Parameter  | Type                                                   | Direction | Description                                                                                                                                                               |
|------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### 18.3.5.13.2 EA\_OnStartValidation

#### Topics:

| Topic   | Detail                                                                                                                              | See Also |
|---------|-------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnStartValidation</b> notifies Add-Ins that a user has invoked the model validation command from Enterprise Architect.</p> |          |

#### Syntax

##### Sub EA\_OnStartValidation(Repository As EA.Repository, ParamArray Args() as Variant)

The *EA\_OnStartValidation* function syntax contains the following elements:

| Parameter  | Type                                                   | Direction | Description                                                                                                    |
|------------|--------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------|
| Args       | <i>ParamArray of Variant</i>                           | IN        | Contains a list of Rule Categories that are active for the current invocation of model validation.             |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to |

| Parameter | Type | Direction | Description                                                |
|-----------|------|-----------|------------------------------------------------------------|
|           |      |           | retrieve model data and user interface status information. |

### 18.3.5.13.3 EA\_OnEndValidation

#### Topics:

| Topic   | Detail                                                                                                                                                         | See Also |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <b>EA_OnEndValidation</b> notifies Add-Ins that model validation has completed. Use this event to arrange any clean-up operations arising from the validation. |          |

#### Syntax

**Sub EA\_OnEndValidation(Repository As EA.Repository, ParamArray Args() as Variant)**

The *EA\_OnEndValidation* function syntax contains the following elements:

| Parameter  | Type                                            | Direction | Description                                                                                                                                                               |
|------------|-------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Args       | <i>ParamArray of Variant</i>                    | IN        | Contains a list of Rule Categories that were active for the invocation of model validation that has just completed.                                                       |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### 18.3.5.13.4 EA\_OnRunElementRule

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See Also                                                  |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| Details | This event is triggered once for each rule defined in <b>EA_OnInitializeUserRules</b> to be performed on each element in the selection being validated. If you don't want to perform the rule defined by <b>RuleID</b> on the given element, then simply return without performing any action. On performing any validation, if a validation error is found, use the <b>Repository.ProjectInterface.PublishResult</b> method to notify Enterprise Architect. Also look at <b>EA_OnInitializeUserRules</b> . | <a href="#">EA_OnInitializeUserRule</a> <sup>[2000]</sup> |

#### Syntax

**Sub EA\_OnRunElementRule(Repository As EA.Repository, RuleID As String, Element As EA.Element)**

The *EA\_OnRunElementRule* function syntax contains the following elements:

| Parameter  | Type                                                   | Direction | Description                                                                                                                                                               |
|------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Element    | <i>EA.Element</i>                                      | IN        | The element to potentially perform validation on.                                                                                                                         |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| RuleID     | <i>String</i>                                          | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                        |

### 18.3.5.13.5 EA\_OnRunPackageRule

Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                       | See Also                                                           |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| Details | This event is triggered once for each rule defined in <b>EA_OnInitializeUserRules</b> to be performed on each package in the selection being validated. If you don't want to perform the rule defined by <b>RuleID</b> on the given package, then simply return without performing any action. On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect. | <a href="#">EA_OnInitializeUserRule</a><br><small>S [2000]</small> |

Syntax

**Sub** *EA\_OnRunPackageRule*(*Repository As EA.Repository, RuleID As String, PackageID As Long*)

The *EA\_OnRunElementRule* function syntax contains the following elements:

| Parameter  | Type                                                   | Direction | Description                                                                                                                                                               |
|------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PackageID  | <i>Long</i>                                            | IN        | The ID of the package to potentially perform validation on. Use the <i>Repository.GetPackageByID</i> method to retrieve the package object.                               |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| RuleID     | <i>String</i>                                          | IN        | The ID that was passed into the <i>Project.DefineRule</i> method.                                                                                                         |

### 18.3.5.13.6 EA\_OnRunDiagramRule

Topics:

| Topic   | Detail                                                                                                        | See Also                                                           |
|---------|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| Details | This event is triggered once for each rule defined in <b>EA_OnInitializeUserRules</b> to be performed on each | <a href="#">EA_OnInitializeUserRule</a><br><small>S [2000]</small> |

| Topic | Detail                                                                                                                                                                                                                                                                                                                                                | See Also |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>diagram in the selection being validated. If you don't want to perform the rule defined by <b>RuleID</b> on the given diagram, then simply return without performing any action. On performing any validation, if a validation error is found, use the <b>Repository.ProjectInterface.PublishResult</b> method to notify Enterprise Architect.</p> |          |

### Syntax

**Sub EA\_OnRunDiagramRule(Repository As EA.Repository, RuleID As String, DiagramID As Long)**

The *EA\_OnRunDiagramRule* function syntax contains the following elements:

| Parameter  | Type                                                   | Direction | Description                                                                                                                                                               |
|------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DiagramID  | Long                                                   | IN        | The ID of the diagram to potentially perform validation on. Use the <i>Repository.GetDiagramByID</i> method to retrieve the diagram object.                               |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| RuleID     | String                                                 | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                        |

### 18.3.5.13.7 EA\_OnRunConnectorRule

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also                                                         |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| Details | <p>This event is triggered once for each rule defined in <b>EA_OnInitializeUserRules</b> to be performed on each connector in the selection being validated. If you don't want to perform the rule defined by <b>RuleID</b> on the given connector, then simply return without performing any action. On performing any validation, if a validation error is found, use the <b>Repository.ProjectInterface.PublishResult</b> method to notify Enterprise Architect.</p> | <a href="#">EA_OnInitializeUserRule</a><br><small>[2000]</small> |

### Syntax

**Sub EA\_OnRunConnectorRule(Repository As EA.Repository, RuleID As String, ConnectorID As Long)**

The *EA\_OnRunConnectorRule* function syntax contains the following elements:

| Parameter   | Type | Direction | Description                                                                                                                     |
|-------------|------|-----------|---------------------------------------------------------------------------------------------------------------------------------|
| ConnectorID | Long | IN        | The ID of the connector to potentially perform validation on. Use the <i>Repository.GetConnectorByID</i> method to retrieve the |

| Parameter  | Type                                    | Direction | Description                                                                                                                                                               |
|------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            |                                         |           | connector object.                                                                                                                                                         |
| Repository | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| RuleID     | String                                  | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                        |

### 18.3.5.13.8 EA\_OnRunAttributeRule

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See Also                                            |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| Details | This event is triggered once for each rule defined in <b>EA_OnInitializeUserRules</b> to be performed on each attribute in the selection being validated. If you don't want to perform the rule defined by <b>RuleID</b> on the given attribute, then simply return without performing any action. On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect. | <a href="#">EA_OnInitializeUserRule</a><br>S [2000] |

#### Syntax

*Sub EA\_OnRunAttributeRule(Repository As EA.Repository, RuleID As String, AttributeGUID As String, ObjectID As Long)*

The *EA\_OnRunAttributeRule* function syntax contains the following elements:

| Parameter     | Type                                    | Direction | Description                                                                                                                                                               |
|---------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AttributeGUID | String                                  | IN        | The GUID of the attribute to potentially perform validation on. Use the <i>Repository.GetAttributeByGuid</i> method to retrieve the attribute object.                     |
| ObjectID      | Long                                    | IN        | The ID of the object that owns the given attribute. Use the <i>Repository.GetObjectByID</i> method to retrieve the object.                                                |
| Repository    | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| RuleID        | String                                  | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                        |

### 18.3.5.13.9 EA\_OnRunMethodRule

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See Also                                                       |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| Details | This event is triggered once for each rule defined in <b>EA_OnInitializeUserRules</b> to be performed on each method in the selection being validated. If you don't want to perform the rule defined by <b>RuleID</b> on the given method, then simply return without performing any action. On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect. | <a href="#">EA_OnInitializeUserRule</a><br>S <sup>[2000]</sup> |

#### Syntax

*Sub EA\_OnRunMethodRule(Repository As EA.Repository, RuleID As String, MethodGUID As String, ObjectID As Long)*

The *EA\_OnRunMethodRule* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                               |
|------------|------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MethodGUID | String                                               | IN        | The GUID of the method to potentially perform validation on. Use the <i>Repository.GetMethodByGuid</i> method to retrieve the method object.                              |
| ObjectID   | Long                                                 | IN        | The ID of the object that owns the given method. Use the <i>Repository.GetObjectByID</i> method to retrieve the object.                                                   |
| Repository | <a href="#">EA.Repository</a><br>S <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| RuleID     | String                                               | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                        |

### 18.3.5.13.10 EA\_OnRunParameterRule

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                           | See Also                                                       |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| Details | This event is triggered once for each rule defined in <b>EA_OnInitializeUserRules</b> to be performed on each parameter in the selection being validated. If you don't want to perform the rule defined by <b>RuleID</b> on the given parameter, then simply return without performing any action. On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect. | <a href="#">EA_OnInitializeUserRule</a><br>S <sup>[2000]</sup> |

#### Syntax

**Sub EA\_OnRunParameterRule(Repository As EA.Repository, RuleID As String, ParameterGUID As String, MethodGUID As String, ObjectID As Long)**

The *EA\_OnRunMethodRule* function syntax contains the following elements:

| Parameter     | Type                                                   | Direction | Description                                                                                                                                                               |
|---------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MethodGUID    | String                                                 | IN        | The GUID of the method that owns the given parameter. Use the <i>Repository.GetMethodByGuid</i> method to retrieve the method object.                                     |
| ObjectID      | Long                                                   | IN        | The ID of the object that owns the given parameter. Use the <i>Repository.GetObjectByID</i> method to retrieve the object.                                                |
| ParameterGUID | String                                                 | IN        | The GUID of the parameter to potentially perform validation on. Use it to retrieve the parameter by iterating through the <i>Method.Parameters</i> collection.            |
| Repository    | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| RuleID        | String                                                 | IN        | The ID that was passed into the <i>Project.DefineRule</i> command.                                                                                                        |

### 18.3.5.13.11 Model Validation Example

The following example code is written in C# and provides a skeleton model validation implementation that you might like to use as a starting point in writing your own model validation rules.

#### Main.cs

```
using System;

namespace myAddIn
{
 public class Main
 {
 public Rules theRules;

 public Main()
 {
 theRules = new Rules();
 }

 public string EA_Connect(EA.Repository Repository)
 {
 return "";
 }

 public void EA_Disconnect()
 {
 GC.Collect();
 GC.WaitForPendingFinalizers();
 }

 private bool IsProjectOpen(EA.Repository Repository)
 {
 try
 {
 EA.Collection c = Repository.Models;
 return true;
 }
 }
 }
}
```

```
 catch
 {
 return false;
 }
 }

 public object EA_GetMenuItems(EA.Repository Repository, string
MenuLocation, string MenuName)
 {
 switch (MenuName)
 {
 case "":
 return "- &myAddin";
 case "- &myAddin":
 string() ar = { "&Test" };
 return ar;
 }
 return "";
 }

 public void EA_GetMenuState(EA.Repository Repository, string
MenuLocation, string MenuName, string ItemName, ref bool IsEnabled, ref
bool IsChecked)
 {
 // if no open project, disable all menu options
 if (IsProjectOpen(Repository))
 IsEnabled = true;
 else
 IsEnabled = false;
 }

 public void EA_MenuClick(EA.Repository Repository, string
MenuLocation, string MenuName, string ItemName)
 {
 switch (ItemName)
 {
 case "&Test";
 DoTest(Repository);
 break;
 }
 }

 public void EA_OnInitializeUserRules(EA.Repository Repository)
 {
 if (Repository != null)
 {
 theRules.ConfigureCategories(Repository);
 theRules.ConfigureRules(Repository);
 }
 }

 public void EA_OnRunElementRule(EA.Repository Repository,
string RuleID, EA.Element element)
 {
 theRules.RunElementRule(Repository, RuleID, element);
 }

 public void EA_OnRunDiagramRule(EA.Repository Repository,
string RuleID, long IDiagramID)
 {
 theRules.RunDiagramRule(Repository, RuleID, IDiagramID);
 }

 public void EA_OnRunConnectorRule(EA.Repository Repository,
string RuleID, long IConnectorID)
 {
 theRules.RunConnectorRule(Repository, RuleID,
IConnectorID);
 }
}
```

```

 public void EA_OnRunAttributeRule(EA.Repository Repository,
string RuleID, string AttGUID, long ObjectId)
 {
 return;
 }

 public void EA_OnDeleteTechnology(EA.Repository Repository, EA.
EventProperties Info)
 {
 return;
 }

 public void EA_OnImportTechnology(EA.Repository Repository, EA.
EventProperties Info)
 {
 return;
 }

 private void DoTest(EA.Repository Rep)
 {
 // TODO: insert test code here
 }
 }
}

```

### Rules.cs

```

using System;
using System.Collections;

namespace myAddIn
{
 public class Rules
 {
 private string m_sCategoryId;
 private System.Collections.ArrayList m_RuleIDs;
 private System.Collections.ArrayList m_RuleDEX;

 private const string cRule01 = "Rule01";
 private const string cRule02 = "Rule02";
 private const string cRule03 = "Rule03";
 // TODO: expand this list as much as necessary

 public Rules()
 {
 m_RuleIDs = new System.Collections.ArrayList();
 m_RuleDEX = new System.Collections.ArrayList();
 }

 private string LookupMap(string sKey)
 {
 return DoLookupMap(sKey, m_RuleIDs, m_RuleDEX);
 }

 private string LookupMapEx(string sRule)
 {
 return DoLookupMap(sRule, m_RuleDEX, m_RuleIDs);
 }

 private string DoLookupMap(string sKey, ArrayList arrValues,
ArrayList arrKeys)
 {
 if (arrKeys.Contains(sKey))
 return arrValues(arrKeys.IndexOf(sKey)).ToString
();
 else
 return "";
 }
 }
}

```

```

private void AddToMap(string sRuleID, string sKey)
{
 m_RuleIDs.Add(sRuleID);
 m_RuleIDEx.Add(sKey);
}

private string GetRuleStr(string sRuleID)
{
 switch (sRuleID)
 {
 case cRule01:
 return "Error Message 01";
 case cRule02:
 return "Error Message 02";
 case cRule03:
 return "Error Message 03";
 // TODO: add extra cases as much as necessary
 }
 return "";
}

public void ConfigureCategories(EA.Repository Repository)
{
 EA.Project Project = Repository.GetProjectInterface();
 m_sCategoryID = Project.DefineRuleCategory("Enterprise
Collaboration Architecture (ECA) Rules");
}

public void ConfigureRules(EA.Repository Repository)
{
 EA.Project Project = Repository.GetProjectInterface();
 AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule01)), cRule01);
 AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule02)), cRule02);
 AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule03)), cRule03);
 // TODO: expand this list
}

public void RunConnectorRule(EA.Repository Repository, string
sRuleID, long lConnectorID)
{
 EA.Connector Connector = Repository.GetConnectorByID
((int)lConnectorID);
 if (Connector != null)
 {
 switch (LookupMapEx(sRuleID))
 {
 case cRule02:
 // TODO: perform rule 2 check
 break;
 // TODO: add more cases
 }
 }
}

public void RunDiagramRule(EA.Repository Repository, string
sRuleID, long lDiagramID)
{
 EA.Diagram Diagram = Repository.GetDiagramByID((int)
lDiagramID);
 if (Diagram != null)
 {
 switch (LookupMapEx(sRuleID))
 {
 case cRule03:
 // TODO: perform rule 3 check
 break;
 // TODO: add more cases
 }
 }
}

```

```

 }
}

public void RunElementRule(EA.Repository Repository, string
sRuleID, EA.Element Element)
{
 if (Element != null)
 {
 switch (LookupMapEx(sRuleID))
 {
 case cRule01:
 DoRule01(Repository, Element);
 break;
 // TODO: add more cases
 }
 }
}

private void DoRule01(EA.Repository Repository, EA.Element
Element)
{
 if (Element.Stereotype != "myStereotype")
 return;

 // TODO: validation logic here

 // report validation errors
 EA.Project Project = Repository.GetProjectInterface();
 Project.PublishResult(LookupMap(cRule01), EA.
EnumMVErrortype.mvError, GetRuleStr(cRule01));
}
}
}

```

### 18.3.5.14 Post-New Events

Enterprise Architect Add-Ins can respond to the creation of new elements, connectors, objects, attributes, methods and packages using the following broadcast events:

| Topic                     | Link                                                        |
|---------------------------|-------------------------------------------------------------|
| EA_OnPostNewElement       | <a href="#">EA_OnPostNewElement</a> <sup>[2011]</sup>       |
| EA_OnPostNewConnector     | <a href="#">EA_OnPostNewConnector</a> <sup>[2011]</sup>     |
| EA_OnPostNewDiagram       | <a href="#">EA_OnPostNewDiagram</a> <sup>[2012]</sup>       |
| EA_OnPostNewDiagramObject | <a href="#">EA_OnPostNewDiagramObject</a> <sup>[2013]</sup> |
| EA_OnPostNewAttribute     | <a href="#">EA_OnPostNewAttribute</a> <sup>[2014]</sup>     |
| EA_OnPostNewMethod        | <a href="#">EA_OnPostNewMethod</a> <sup>[2015]</sup>        |
| EA_OnPostNewPackage       | <a href="#">EA_OnPostNewPackage</a> <sup>[2015]</sup>       |
| EA_OnPostNewGlossaryTerm. | <a href="#">EA_OnPostNewGlossaryTerm</a> <sup>[2016]</sup>  |

### 18.3.5.14.1 EA\_OnPostNewElement

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See Also                                             |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| Details | <p><b>EA_OnPostNewElement</b> notifies Add-Ins that a new element has been created on a diagram. It enables Add-Ins to modify the element upon creation.</p> <p>This event occurs after a user has dragged a new element from the <b>Toolbox</b> or <b>Resources</b> window onto a diagram. The notification is provided immediately after the element is added to the model. Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> <p>Also look at <b>EA_OnPreNewElement</b>.</p> | <a href="#">EA_OnPreNewElement</a> <sup>[2024]</sup> |

#### Syntax

**Function** EA\_OnPostNewElement(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewElement* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                                                          |
|------------|------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the new element: <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>Element.ElementID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                            |

#### Return Value

Return **True** if the element has been updated during this notification. Return **False** otherwise.

### 18.3.5.14.2 EA\_OnPostNewConnector

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                  | See Also                                               |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| Details | <p><b>EA_OnPostNewConnector</b> notifies Add-Ins that a new connector has been created on a diagram. It enables Add-Ins to modify the connector upon creation.</p> <p>This event occurs after a user has dragged a new connector from the <b>Toolbox</b> or <b>Resources</b> window onto a diagram. The notification is provided immediately after the connector is</p> | <a href="#">EA_OnPreNewConnector</a> <sup>[2025]</sup> |

| Topic | Detail                                                                                                                                                                                                | See Also |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>added to the model. Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> <p>Also look at <b>EA_OnPreNewConnector</b>.</p> |          |

### Syntax

**Function** *EA\_OnPostNewConnector*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewConnector* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                         |
|-------------------|------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the new connector:</p> <ul style="list-style-type: none"> <li><i>ConnectorID</i>: A long value corresponding to <i>Connector.ConnectorID</i>.</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                           |

### Return Value

Return **True** if the connector has been updated during this notification. Return **False** otherwise.

#### 18.3.5.14.3 EA\_OnPostNewDiagram

### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                   | See Also                                             |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| <b>Details</b> | <p><b>EA_OnPostNewDiagram</b> notifies Add-Ins that a new diagram has been created. It enables Add-Ins to modify the diagram upon creation.</p> <p>Set <i>Repository.SuppressEADialogs</i> to true to suppress Enterprise Architect from showing its default dialogs.</p> <p>Also look at <b>EA_OnPreNewDiagram</b>.</p> | <a href="#">EA_OnPreNewDiagram</a> <sup>[2028]</sup> |

### Syntax

**Function** *EA\_OnPostNewDiagram*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewDiagram* function syntax contains the following elements:

| Parameter  | Type                                                        | Direction | Description                                                                                                                                                                                          |
|------------|-------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects for the new diagram: <ul style="list-style-type: none"> <li><i>DiagramID</i>: A long value corresponding to <i>Diagram.PackageID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                            |

### Return Value

Return **True** if the diagram has been updated during this notification. Return **False** otherwise.

### 18.3.5.14.4 EA\_OnPostNewDiagramObject

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also                                                          |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| Details | <p><i>EA_OnPostNewDiagramObject</i> notifies Add-Ins that a new object has been created on a diagram. It enables Add-Ins to modify the object upon creation.</p> <p>This event occurs after a user has dragged a new object from the <b>Project Browser</b> or <b>Resources</b> window onto a diagram. The notification is provided immediately after the object is added to the diagram. <i>Set Repository</i>. <i>SuppressEADialogs</i> to true to suppress Enterprise Architect from showing its default dialogs.</p> <p>Also look at <b>EA_OnPreNewDiagramObject</b>.</p> | <a href="#">EA_OnPreNewDiagramObject</a><br><small>[2028]</small> |

### Syntax

**Function** *EA\_OnPostNewDiagramObject*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewDiagramObject* function syntax contains the following elements:

| Parameter  | Type                                                        | Direction | Description                                                                                                                                                                                       |
|------------|-------------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects for the new element: <ul style="list-style-type: none"> <li><i>ObjectID</i>: A long value corresponding to <i>Object.ObjectID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                         |

### Return Value

Return **True** if the element has been updated during this notification. Return **False** otherwise.

#### 18.3.5.14.5 EA\_OnPostNewAttribute

##### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | See Also                                       |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewAttribute</i> notifies Add-Ins that a new attribute has been created on a diagram. It enables Add-Ins to modify the attribute upon creation.</p> <p>This event occurs when a user creates a new attribute on an element by either drag-dropping from the <b>Project Browser</b>, using the <b>Attributes Properties</b> dialog, or using the in-place editor on the diagram. The notification is provided immediately after the attribute is created. Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> <p>Also look at <b>EA_OnPreNewAttribute</b>.</p> | <a href="#">EA_OnPreNewAttribute</a><br>[2027] |

##### Syntax

**Function** EA\_OnPostNewAttribute(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewAttribute* function syntax contains the following elements:

| Parameter         | Type                                         | Direction | Description                                                                                                                                                                                                  |
|-------------------|----------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br>[1844] | IN        | Contains the following <i>EventProperty</i> objects for the new attribute: <ul style="list-style-type: none"> <li><i>AttributeID</i>: A long value corresponding to <i>Attribute.AttributeID</i>.</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[1855]      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                    |

##### Return Value

Return **True** if the attribute has been updated during this notification. Return **False** otherwise.

### 18.3.5.14.6 EA\_OnPostNewMethod

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See Also                                    |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| Details | <p><b>EA_OnPostNewMethod</b> notifies Add-Ins that a new method has been created on a diagram. It enables Add-Ins to modify the method upon creation.</p> <p>This event occurs when a user creates a new method on an element by either drag-dropping from the <b>Project Browser</b>, using the method's <b>Properties</b> dialog, or using the in-place editor on the diagram. The notification is provided immediately after the method is created. Set <i>Repository.SuppressEADialogs</i> to true to suppress Enterprise Architect from showing its default dialogs.</p> <p>Also look at <b>EA_OnPreNewMethod</b>.</p> | <a href="#">EA_OnPreNewMethod</a><br>[2028] |

#### Syntax

**Function** EA\_OnPostNewMethod(*Repository As EA.Repository, Info As EA.EventProperties*) As Boolean

The *EA\_OnPostNewMethod* function syntax contains the following elements:

| Parameter  | Type                                         | Direction | Description                                                                                                                                                                                      |
|------------|----------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br>[1844] | IN        | Contains the following <i>EventProperty</i> objects for the new method: <ul style="list-style-type: none"> <li><i>MethodID</i>: A long value corresponding to <i>Method.MethodID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br>[1855]      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                        |

#### Return Value

Return **True** if the method has been updated during this notification. Return **False** otherwise.

### 18.3.5.14.7 EA\_OnPostNewPackage

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                | See Also                                     |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| Details | <p><b>EA_OnPostNewPackage</b> notifies Add-Ins that a new package has been created on a diagram. It enables Add-Ins to modify the package upon creation.</p> <p>This event occurs when a user drags a new package from the <b>Toolbox</b> or <b>Resources</b> window onto a diagram, or by selecting the <b>New Package</b> icon from the <b>Project Browser</b>.</p> | <a href="#">EA_OnPreNewPackage</a><br>[2029] |

| Topic | Detail                                                                                                                                                                          | See Also |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> <p>Also look at <b>EA_OnPreNewPackage</b>.</p> |          |

### Syntax

**Function** EA\_OnPostNewPackage(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewPackage* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                 |
|-------------------|-------------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | <p>Contains the following <i>EventProperty</i> objects for the new package:</p> <ul style="list-style-type: none"> <li><i>PackageID</i>: A long value corresponding to <i>Package.PackageID</i>.</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                   |

### Return Value

Return **True** if the package has been updated during this notification. Return **False** otherwise.

#### 18.3.5.14.8 EA\_OnPostNewGlossaryTerm

### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also                                                         |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>Details</b> | <p><i>EA_OnPostNewGlossaryTerm</i> notifies Add-Ins that a new glossary term has been created. It enables Add-Ins to modify the glossary term upon creation.</p> <p>The notification is provided immediately after the glossary term is added to the model. Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default dialogs.</p> <p>Also look at <b>EA_OnPreNewGlossaryTerm</b>.</p> | <a href="#">EA_OnPreNewGlossaryTerm</a><br><small>[2030]</small> |

### Syntax

**Function** EA\_OnPostNewGlossaryTerm(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewGlossaryTerm* function syntax contains the following elements:

| Parameter         | Type                                                 | Direction | Description                                                                                                                                                                                                |
|-------------------|------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the new glossary term: <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>Element.ElementID</i>.</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                  |

### Return Value

Return **True** if the glossary term has been updated during this notification. Return **False** otherwise.

### 18.3.5.15 Pre-Deletion Events

Enterprise Architect Add-Ins can respond to requests to delete elements, attributes, methods, connectors, diagrams, packages and glossary terms using the following broadcast events:

| Topic                                 | Link                                                         |
|---------------------------------------|--------------------------------------------------------------|
| EA_OnPreDeleteElement                 | <a href="#">EA_OnPreDeleteElement</a> <sup>[2017]</sup>      |
| EA_OnPreDeleteAttribute               | <a href="#">EA_OnPreDeleteAttribute</a> <sup>[2018]</sup>    |
| EA_OnPreDeleteMethod                  | <a href="#">EA_OnPreDeleteMethod</a> <sup>[2019]</sup>       |
| EA_OnPreDeleteConnector               | <a href="#">EA_OnPreDeleteConnector</a> <sup>[2020]</sup>    |
| EA_OnPreDeleteDiagram                 | <a href="#">EA_OnPreDeleteDiagram</a> <sup>[2020]</sup>      |
| EA_OnPreDeletePackage                 | <a href="#">EA_OnPreDeletePackage</a> <sup>[2022]</sup>      |
| EA_OnPreDeleteGlossaryTerm            | <a href="#">EA_OnPreDeleteGlossaryTerm</a> <sup>[2023]</sup> |
| EA_OnPreDeleteTechnology (Deprecated) | <a href="#">EA_OnPreDeleteTechnology</a> <sup>[2037]</sup>   |

#### 18.3.5.15.1 EA\_OnPreDeleteElement

##### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                       | See Also |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><b>EA_OnPreDeleteElement</b> notifies Add-Ins that an element is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the element.</p> <p>This event occurs when a user deletes an element from the <b>Project Browser</b> or on a diagram. The notification is provided immediately before the element is deleted, so that the Add-In can disable deletion of the element.</p> |          |

### Syntax

**Function** EA\_OnPreDeleteElement(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteElement* function syntax contains the following elements:

| Parameter  | Type                                                        | Direction | Description                                                                                                                                                                                                    |
|------------|-------------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects for the element to be deleted: <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>Element.ElementID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                      |

### Return Value

Return **True** to enable deletion of the element from the model. Return **False** to disable deletion of the element.

#### 18.3.5.15.2 EA\_OnPreDeleteAttribute

### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                          | See Also |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnPreDeleteAttribute</b> notifies Add-Ins that an attribute is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the attribute.</p> <p>This event occurs when a user attempts to permanently delete an attribute from the <b>Project Browser</b>. The notification is provided immediately before the attribute is deleted, so that the Add-In can disable deletion of the attribute.</p> |          |

### Syntax

**Function** EA\_OnPreDeleteAttribute(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteAttribute* function syntax contains the following elements:

| Parameter  | Type                                                        | Direction | Description                                                                                                                                                                                                            |
|------------|-------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects for the attribute to be deleted: <ul style="list-style-type: none"> <li><i>AttributeID</i>: A long value corresponding to <i>Attribute.AttributeID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open                                                                                                                                                         |

| Parameter | Type | Direction | Description                                                                                                |
|-----------|------|-----------|------------------------------------------------------------------------------------------------------------|
|           |      |           | Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

Return **True** to enable deletion of the attribute from the model. Return **False** to disable deletion of the attribute.

### 18.3.5.15.3 EA\_OnPreDeleteMethod

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnPreDeleteMethod</b> notifies Add-Ins that a method (operation) is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the method.</p> <p>This event occurs when a user attempts to permanently delete a method from the <b>Project Browser</b>. The notification is provided immediately before the method is deleted, so that the Add-In can disable deletion of the method.</p> |          |

#### Syntax

**Function** EA\_OnPreDeleteMethod(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteMethod* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                |
|-------------------|-------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects for the method to be deleted: <ul style="list-style-type: none"> <li><i>MethodID</i>: A long value corresponding to <i>Method.MethodID</i>.</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                  |

### Return Value

Return **True** to enable deletion of the method from the model. Return **False** to disable deletion of the method.

#### 18.3.5.15.4 EA\_OnPreDeleteConnector

##### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                     | See Also |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnPreDeleteConnector</b> notifies Add-Ins that a connector is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the connector.</p> <p>This event occurs when a user attempts to permanently delete a connector on a diagram. The notification is provided immediately before the connector is deleted, so that the Add-In can disable deletion of the connector.</p> |          |

##### Syntax

**Function** EA\_OnPreDeleteConnector(*Repository As EA.Repository, Info As EA.EventProperties*) As Boolean

The *EA\_OnPreDeleteConnector* function syntax contains the following elements:

| Parameter  | Type                                                        | Direction | Description                                                                                                                                                                                                            |
|------------|-------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects for the connector to be deleted: <ul style="list-style-type: none"> <li><i>ConnectorID</i>: A long value corresponding to <i>Connector.ConnectorID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                              |

##### Return Value

Return **True** to enable deletion of the connector from the model. Return **False** to disable deletion of the connector.

#### 18.3.5.15.5 EA\_OnPreDeleteDiagram

##### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                            | See Also |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnPreDeleteDiagram</b> notifies Add-Ins that a diagram is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the diagram.</p> <p>This event occurs when a user attempts to permanently delete a diagram from the <b>Project Browser</b>. The notification is provided immediately before the diagram is deleted, so that the Add-In can disable deletion of the diagram.</p> |          |

### Syntax

**Function** *EA\_OnPreDeleteDiagram(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreDeleteDiagram* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects for the diagram to be deleted: <ul style="list-style-type: none"> <li><i>DiagramID</i>: A long value corresponding to <i>Diagram.DiagramID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently-open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                     |

### Return Value

Return **True** to enable deletion of the diagram from the model. Return **False** to disable deletion of the diagram.

#### 18.3.5.15.6 EA\_OnPreDeleteDiagramObject

### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                         | See Also |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Details</b> | <p><b>EA_OnPreDeleteDiagramObject</b> notifies Add-Ins that a diagram object is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the element.</p> <p>This event occurs when a user attempts to permanently delete an element from a diagram. The notification is provided immediately before the element is deleted, so that the Add-In can disable deletion of the element.</p> |          |

### Syntax

**Function** *EA\_OnPreDeleteDiagramObject(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreDeleteDiagramObject* function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                  |
|-------------------|-------------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects for the element to be deleted: <ul style="list-style-type: none"> <li><i>ID</i>: A long value corresponding to <i>DiagramObject.ElementID</i></li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a>                               | IN        | An <i>EA.Repository</i> object representing the currently-open                                                                                                                                               |

| Parameter | Type                                                                                   | Direction | Description                                                                                                |
|-----------|----------------------------------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------|
| y         |  1855 |           | Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

Return **True** to enable deletion of the element from the model. Return **False** to disable deletion of the element.

### 18.3.5.15.7 EA\_OnPreDeletePackage

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                            | See Also |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnPreDeletePackage</b> notifies Add-Ins that a package is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the package.</p> <p>This event occurs when a user attempts to permanently delete a package from the <b>Project Browser</b>. The notification is provided immediately before the package is deleted, so that the Add-In can disable deletion of the package.</p> |          |

#### Syntax

**Function** EA\_OnPreDeletePackage(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeletePackage* function syntax contains the following elements:

| Parameter  | Type                                                                                                                           | Direction | Description                                                                                                                                                                                                    |
|------------|--------------------------------------------------------------------------------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br> 1844 | IN        | Contains the following <i>EventProperty</i> objects for the package to be deleted: <ul style="list-style-type: none"> <li><i>PackageID</i>: A long value corresponding to <i>Package.PackageID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br> 1855      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                      |

### Return Value

Return **True** to enable deletion of the package from the model. Return **False** to disable deletion of the package.

### 18.3.5.15.8 EA\_OnPreDeleteGlossaryTerm

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                 | See Also |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnPreDeleteGlossaryTerm</b> notifies Add-Ins that a glossary term is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the glossary term.</p> <p>The notification is provided immediately before the glossary term is deleted, so that the Add-In can disable deletion of the glossary term.</p> |          |

#### Syntax

**Function** EA\_OnPreDeleteGlossaryTerm(*Repository As EA.Repository, Info As EA.EventProperties*) As Boolean

The *EA\_OnPreDeleteGlossaryTerm* function syntax contains the following elements:

| Parameter  | Type                                         | Direction | Description                                                                                                                                                                                                 |
|------------|----------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br>[1844] | IN        | Contains the following <i>EventProperty</i> objects for the glossary term to be deleted: <ul style="list-style-type: none"> <li><i>TermID</i>: A long value corresponding to <i>Term.TermID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br>[1855]      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                   |

#### Return Value

Return **True** to enable deletion of the glossary term from the model. Return **False** to disable deletion of the glossary term.

### 18.3.5.16 Pre-New Events

Enterprise Architect Add-Ins can respond to requests to create new elements, connectors, objects, attributes, methods and packages using the following broadcast events:

| Topic                    | Link                                               |
|--------------------------|----------------------------------------------------|
| EA_OnPreNewElement       | <a href="#">EA_OnPreNewElement</a><br>[2024]       |
| EA_OnPreNewConnector     | <a href="#">EA_OnPreNewConnector</a><br>[2025]     |
| EA_OnPreNewDiagram       | <a href="#">EA_OnPreNewDiagram</a><br>[2026]       |
| EA_OnPreNewDiagramObject | <a href="#">EA_OnPreNewDiagramObject</a><br>[2026] |
| EA_OnPreNewAttribute     | <a href="#">EA_OnPreNewAttribute</a><br>[2027]     |

| Topic                    | Link                                                      |
|--------------------------|-----------------------------------------------------------|
| EA_OnPreNewMethod        | <a href="#">EA_OnPreNewMethod</a> <sup>[2028]</sup>       |
| EA_OnPreNewPackage       | <a href="#">EA_OnPreNewPackage</a> <sup>[2029]</sup>      |
| EA_OnPreNewGlossaryTerm. | <a href="#">EA_OnPreNewGlossaryTerm</a> <sup>[2030]</sup> |

### 18.3.5.16.1 EA\_OnPreNewElement

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | See Also                                              |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| Details | <p><b>EA_OnPreNewElement</b> notifies Add-Ins that a new element is about to be created on a diagram. It enables Add-Ins to permit or deny creation of the new element.</p> <p>This event occurs when a user drags a new element from the <b>Toolbox</b> or <b>Resources</b> window onto a diagram. The notification is provided immediately before the element is created, so that the Add-In can disable addition of the element.</p> <p>Also look at <b>EA_OnPostNewElement</b>.</p> | <a href="#">EA_OnPostNewElement</a> <sup>[2017]</sup> |

#### Syntax

**Function** EA\_OnPreNewElement(*Repository As EA.Repository, Info As EA.EventProperties*) As Boolean

The *EA\_OnPreNewElement* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the element to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Element.Type</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Element.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Element.ParentID</i></li> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the element is being added.</li> </ul> |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                              |

#### Return Value

Return **True** to enable addition of the new element to the model. Return **False** to disable addition of the new element.

### 18.3.5.16.2 EA\_OnPreNewConnector

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See Also                                                |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| Details | <p><b>EA_OnPreNewConnector</b> notifies Add-Ins that a new connector is about to be created on a diagram. It enables Add-Ins to permit or deny creation of a new connector.</p> <p>This event occurs when a user drags a new connector from the <b>Toolbox</b> or <b>Resources</b> window, onto a diagram. The notification is provided immediately before the connector is created, so that the Add-In can disable addition of the connector.</p> <p>Also look at <b>EA_OnPostNewConnector</b>.</p> | <a href="#">EA_OnPostNewConnector</a> <sup>[2011]</sup> |

#### Syntax

**Function** EA\_OnPreNewConnector(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreNewConnector* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------|------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | <p>Contains the following <i>EventProperty</i> objects for the connector to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Connector.Type</i></li> <li><i>Subtype</i>: A string value corresponding to <i>Connector.Subtype</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Connector.Stereotype</i></li> <li><i>ClientID</i>: A long value corresponding to <i>Connector.ClientID</i></li> <li><i>SupplierID</i>: A long value corresponding to <i>Connector.SupplierID</i></li> <li><i>DiagramID</i>: A long value corresponding to <i>Connector.DiagramID</i></li> </ul> |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

#### Return Value

Return **True** to enable addition of the new connector to the model. Return **False** to disable addition of the new connector.

### 18.3.5.16.3 EA\_OnPreNewDiagram

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                | See Also                                              |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| Details | <p><b>EA_OnPreNewDiagram</b> notifies Add-Ins that a new diagram is about to be created. It enables Add-Ins to permit or deny creation of the new diagram.</p> <p>The notification is provided immediately before the diagram is created, so that the Add-In can disable addition of the diagram.</p> <p>Also look at <b>EA_OnPostNewDiagram</b>.</p> | <a href="#">EA_OnPostNewDiagram</a> <sup>[2012]</sup> |

#### Syntax

**Function** EA\_OnPreNewDiagram(*Repository As EA.Repository, Info As EA.EventProperties*) As Boolean

The *EA\_OnPreNewDiagram* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                              |
|------------|------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the diagram to be created: <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Diagram.Type</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Diagram.ParentID</i></li> <li><i>PackageID</i>: A long value corresponding to <i>Diagram.PackageID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                |

#### Return Value

Return **True** to enable addition of the new diagram to the model. Return **False** to disable addition of the new diagram.

### 18.3.5.16.4 EA\_OnPreNewDiagramObject

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                | See Also                                                    |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| Details | <p><b>EA_OnPreNewDiagramObject</b> notifies Add-Ins that a new diagram object is about to be dropped on a diagram. It enables Add-Ins to permit or deny creation of the new object.</p> <p>This event occurs when a user drags an object from the Enterprise Architect <b>Project Browser</b> or <b>Resources</b> window onto a diagram. The notification is provided immediately</p> | <a href="#">EA_OnPostNewDiagramObject</a> <sup>[2013]</sup> |

| Topic | Detail                                                                                                                                            | See Also |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p>before the object is created, so that the Add-In can disable addition of the object.</p> <p>Also look at <b>EA_OnPostNewDiagramObject</b>.</p> |          |

### Syntax

**Function** `EA_OnPreNewDiagramObject(Repository As EA.Repository, Info As EA.EventProperties) As Boolean`

The `EA_OnPreNewDiagramObject` function syntax contains the following elements:

| Parameter         | Type                                                        | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | <p>Contains the following <i>EventProperty</i> objects for the object to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Object.Type</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Object.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Object.ParentID</i></li> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the object is being added.</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                                         |

### Return Value

Return **True** to enable addition of the object to the model. Return **False** to disable addition of the object.

#### 18.3.5.16.5 EA\_OnPreNewAttribute

### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also                                                       |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
| <b>Details</b> | <p><b>EA_OnPreNewAttribute</b> notifies Add-Ins that a new attribute is about to be created on an element. It enables Add-Ins to permit or deny creation of the new attribute.</p> <p>This event occurs when a user creates a new attribute on an element by either drag-dropping from the <b>Project Browser</b>, using the <b>Attributes Properties</b> dialog, or using the in-place editor on the diagram. The notification is provided immediately before the attribute is created, so that the Add-In can disable addition of the attribute.</p> <p>Also look at <b>EA_OnPostNewAttribute</b>.</p> | <a href="#">EA_OnPostNewAttribute</a><br><small>[2014]</small> |

### Syntax

**Function** *EA\_OnPreNewAttribute(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewAttribute* function syntax contains the following elements:

| Parameter         | Type                                         | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b>       | <a href="#">EA.EventProperties</a><br>[1844] | IN        | Contains the following <i>EventProperty</i> objects for the attribute to be created: <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Attribute.Type</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Attribute.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Attribute.ParentID</i></li> <li><i>ClassifierID</i>: A long value corresponding to <i>Attribute.ClassifierID</i>.</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br>[1855]      | IN        | An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                                                             |

#### Return Value

Return **True** to enable addition of the new attribute to the model. Return **False** to disable addition of the new attribute.

#### 18.3.5.16.6 *EA\_OnPreNewMethod*

#### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See Also                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| <b>Details</b> | <p><b>EA_OnPreNewMethod</b> notifies Add-Ins that a new method is about to be created on an element. It enables Add-Ins to permit or deny creation of the new method.</p> <p>This event occurs when a user creates a new method on an element by either drag-dropping from the <b>Project Browser</b>, using the method <b>Properties</b> dialog, or using the in-place editor on the diagram. The notification is provided immediately before the method is created, so that the Add-In can disable addition of the method.</p> <p>Also look at <b>EA_OnPostNewMethod</b>.</p> | <a href="#">EA_OnPostNewMethod</a><br>[2015] |

#### Syntax

**Function** *EA\_OnPreNewMethod(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewMethod* function syntax contains the following elements:

| Parameter   | Type                               | Direction | Description                                                                       |
|-------------|------------------------------------|-----------|-----------------------------------------------------------------------------------|
| <b>Info</b> | <a href="#">EA.EventProperties</a> | IN        | Contains the following <i>EventProperty</i> objects for the method to be created: |

| Parameter         | Type                                                                                                                      | Direction | Description                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   |  [1844]                                  |           | <ul style="list-style-type: none"> <li><i>ReturnType</i>: A string value corresponding to <i>Method.ReturnType</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Method.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Method.ParentID</i></li> <li><i>ClassifierID</i>: A long value corresponding to <i>Method.ClassifierID</i>.</li> </ul> |
| <b>Repository</b> | <a href="#">EA.Repository</a><br> [1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                 |

### Return Value

Return **True** to enable addition of the new method to the model. Return **False** to disable addition of the new method.

### 18.3.5.16.7 EA\_OnPreNewPackage

#### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See Also                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><b>EA_OnPreNewPackage</b> notifies Add-Ins that a new package is about to be created in the model. It enables Add-Ins to permit or deny creation of the new package.</p> <p>This event occurs when a user drags a new package from the <b>Toolbox</b> or <b>Resources</b> window onto a diagram, or by selecting the <b>New Package</b> icon from the <b>Project Browser</b>. The notification is provided immediately before the package is created, so that the Add-In can disable addition of the package.</p> <p>Also look at <b>EA_OnPostNewPackage</b>.</p> | <a href="#">EA_OnPostNewPackage</a><br> [2015] |

### Syntax

**Function** EA\_OnPreNewPackage(*Repository* As EA.Repository, *Info* As EA.EventProperties) As Boolean

The *EA\_OnPreNewPackage* function syntax contains the following elements:

| Parameter   | Type                                                                                                                             | Direction | Description                                                                                                                                                                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Info</b> | <a href="#">EA.EventProperties</a><br> [1843] | IN        | <p>Contains the following <i>EventProperty</i> objects for the package to be created:</p> <ul style="list-style-type: none"> <li><i>Stereotype</i>: A string value corresponding to <i>Package.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Package.ParentID</i></li> </ul> |

| Parameter  | Type                                                   | Direction | Description                                                                                                                                                               |
|------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            |                                                        |           | <ul style="list-style-type: none"> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the package is being added.</li> </ul>              |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

Return **True** to enable addition of the new package to the model. Return **False** to disable addition of the new package.

### 18.3.5.16.8 EA\_OnPreNewGlossaryTerm

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                               | See Also                                                          |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| Details | <p><b>EA_OnPreNewGlossaryTerm</b> notifies Add-Ins that a new glossary term is about to be created. It enables Add-Ins to permit or deny creation of the new glossary term.</p> <p>The notification is provided immediately before the glossary term is created, so that the Add-In can disable addition of the element.</p> <p>Also look at <a href="#">EA_OnPostNewGlossaryTerm</a><br/><small>[2016]</small>.</p> | <a href="#">EA_OnPostNewGlossaryTerm</a><br><small>[2016]</small> |

### Syntax

**Function** *EA\_OnPreNewGlossaryTerm(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewGlossaryTerm* function syntax contains the following elements:

| Parameter  | Type                                                        | Direction | Description                                                                                                                                                                                                |
|------------|-------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> object for the glossary term to be created: <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Term.TermID</i>.</li> </ul> |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                  |

### Return Value

Return **True** to enable addition of the new glossary term to the model. Return **False** to disable addition of the new glossary term.

### 18.3.5.17 Tagged Value Broadcasts

Enterprise Architect includes the *Addin Broadcast* Tagged Value type that allows an Add-In to respond to attempts to edit it. The function that is called depends on the type of object the Tagged Value is on.

| Topic                 | Link                                                    |
|-----------------------|---------------------------------------------------------|
| EA_OnAttributeTagEdit | <a href="#">EA_OnAttributeTagEdit</a> <sup>[2031]</sup> |
| EA_OnConnectorTagEdit | <a href="#">EA_OnConnectorTagEdit</a> <sup>[2032]</sup> |
| EA_OnElementTagEdit   | <a href="#">EA_OnElementTagEdit</a> <sup>[2033]</sup>   |
| EA_OnOperationTagEdit | <a href="#">EA_OnOperationTagEdit</a> <sup>[2034]</sup> |

#### 18.3.5.17.1 EA\_OnAttributeTagEdit

##### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                     | See Also                                                                                                                                                                                                                                                                                                                  |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><i>EA_OnAttributeTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type <i>AddinBroadcast</i> on an attribute</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function</p> | <p><a href="#">Predefined Structured Types</a><sup>[1108]</sup></p> <p><a href="#">EA_OnAttributeTagEdit</a><sup>[2031]</sup></p> <p><a href="#">EA_OnConnectorTagEdit</a><sup>[2032]</sup></p> <p><a href="#">EA_OnElementTagEdit</a><sup>[2033]</sup></p> <p><a href="#">EA_OnOperationTagEdit</a><sup>[2034]</sup></p> |

##### Syntax

**Sub EA\_OnAttributeTagEdit(Repository As EA.Repository, ObjectID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnAttributeTagEdit* function syntax contains the following elements:

| Parameter  | Type                                            | Direction | Description                                                                                                                                                                |
|------------|-------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ObjectID   | Long                                            | IN        | The ID of the object (Attribute) that this Tagged Value is on                                                                                                              |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model<br>Poll its members to retrieve model data and user interface status information |
| TagName    | String                                          | IN        | The name of the Tagged Value to edit                                                                                                                                       |
| TagNotes   | String                                          | INOUT     | The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function                                    |

| Parameter | Type   | Direction | Description                                                                                                              |
|-----------|--------|-----------|--------------------------------------------------------------------------------------------------------------------------|
| TagValue  | String | INOUT     | The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function |

### 18.3.5.17.2 EA\_OnConnectorTagEdit

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                             | See Also                                                                                                                                                                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><i>EA_OnConnectorTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type AddinBroadcast on a connector</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function</p> | <p><a href="#">Predefined Structured Types</a> <sup>[1108]</sup></p> <p><a href="#">EA_OnAttributeTagEdit</a> <sub>[2031]</sub></p> <p><a href="#">EA_OnElementTagEdit</a> <sub>[2033]</sub></p> <p><a href="#">EA_OnOperationTagEdit</a> <sub>[2034]</sub></p> |

#### Syntax

**Sub** EA\_OnConnectorTagEdit(*Repository* As EA.Repository, *ObjectID* As Long, *String* TagName, *String* TagValue, *String* TagNotes)

The *EA\_OnConnectorTagEdit* function syntax contains the following elements:

| Parameter  | Type                                            | Direction | Description                                                                                                                                                                |
|------------|-------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ObjectID   | Long                                            | IN        | The ID of the object (connector) that this Tagged Value is on                                                                                                              |
| Repository | <a href="#">EA.Repository</a> <sub>[1855]</sub> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model<br>Poll its members to retrieve model data and user interface status information |
| TagName    | String                                          | IN        | The name of the Tagged Value to edit                                                                                                                                       |
| TagNotes   | String                                          | INOUT     | The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function                                    |
| TagValue   | String                                          | INOUT     | The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function                                                   |

### 18.3.5.17.3 EA\_OnElementTagEdit

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                         | See Also                                                                                                                                                                                                                                                          |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><i>EA_OnElementTagEdit</i> is called when the user clicks the ellipsis ( ...) button for a Tagged Value of type AddinBroadcast on an element</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function</p> | <p><a href="#">Predefined Structured Types</a> <sup>[1108]</sup></p> <p><a href="#">EA_OnAttributeTagEdit</a> <sup>[2031]</sup></p> <p><a href="#">EA_OnConnectorTagEdit</a> <sup>[2032]</sup></p> <p><a href="#">EA_OnOperationTagEdit</a> <sup>[2034]</sup></p> |

#### Syntax

**Sub EA\_OnElementTagEdit(Repository As EA.Repository, ObjectID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnElementTagEdit* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                |
|-------------------|-------------------------------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ObjectID</b>   | Long                                            | IN        | The ID of the object (element) that this Tagged Value is on                                                                                                                |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model<br>Poll its members to retrieve model data and user interface status information |
| <b>TagName</b>    | String                                          | IN        | The name of the Tagged Value to edit                                                                                                                                       |
| <b>TagNotes</b>   | String                                          | INOUT     | The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function                                    |
| <b>TagValue</b>   | String                                          | INOUT     | The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function                                                   |

### 18.3.5.17.4 EA\_OnOperationTagEdit

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                     | See Also                                                                                                                                                                                                                                                        |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><i>EA_OnOperationTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type <i>AddinBroadcast</i> on an operation</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function</p> | <p><a href="#">Predefined Structured Types</a> <sup>[1108]</sup></p> <p><a href="#">EA_OnAttributeTagEdit</a> <sup>[2031]</sup></p> <p><a href="#">EA_OnConnectorTagEdit</a> <sup>[2032]</sup></p> <p><a href="#">EA_OnElementTagEdit</a> <sup>[2033]</sup></p> |

#### Syntax

**Sub EA\_OnOperationTagEdit(Repository As EA.Repository, ObjectID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnOperationTagEdit* function syntax contains the following elements:

| Parameter         | Type                                            | Direction | Description                                                                                                                                                                    |
|-------------------|-------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ObjectID</b>   | Long                                            | IN        | The ID of the object (operation) that this Tagged Value is on                                                                                                                  |
| <b>Repository</b> | <a href="#">EA.Repository</a> <sup>[1855]</sup> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model<br><br>Poll its members to retrieve model data and user interface status information |
| <b>TagName</b>    | String                                          | IN        | The name of the Tagged Value to edit                                                                                                                                           |
| <b>TagNotes</b>   | String                                          | INOUT     | The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function                                        |
| <b>TagValue</b>   | String                                          | INOUT     | The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function                                                       |

### 18.3.5.18 Technology Events

Enterprise Architect Add-Ins can respond to the following events associated with the use of MDG Technologies:

| Topic                       | Link                                                          |
|-----------------------------|---------------------------------------------------------------|
| EA_OnInitializeTechnologies | <a href="#">EA_OnInitializeTechnologies</a> <sup>[2035]</sup> |
| EA_OnPreActivateTechnology  | <a href="#">EA_OnPreActivateTechnology</a> <sup>[2036]</sup>  |
| EA_OnPostActivateTechnology | <a href="#">EA_OnPostActivateTechnology</a>                   |

| Topic                                 | Link                                                                                                                                   |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
|                                       |  [2037]                                             |
| EA_OnPreDeleteTechnology (Deprecated) | <a href="#">EA_OnPreDeleteTechnology</a><br> [2037] |
| EA_OnDeleteTechnology (Deprecated)    | <a href="#">EA_OnDeleteTechnology</a>  [2038]       |
| EA_OnImportTechnology (Deprecated)    | <a href="#">EA_OnImportTechnology</a>  [2039]       |

### 18.3.5.18.1 EA\_OnInitializeTechnologies

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                             | See Also |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnInitializeTechnologies</b> requests that an Add-In pass an MDG Technology to Enterprise Architect for loading.</p> <p>This event occurs on Enterprise Architect startup. Return your technology XML to this function and Enterprise Architect loads and enables it.</p> |          |

#### Syntax

**Function** EA\_OnInitializeTechnologies(*Repository* As EA.Repository) As Object

The *EA\_OnInitializeTechnologies* function syntax contains the following element:

| Parameter  | Type                                                                                                                        | Direction | Description                                                                                                                                                               |
|------------|-----------------------------------------------------------------------------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a><br> [1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

Return the MDG Technology as a single XML string.

#### Example

```
Public Function EA_OnInitializeTechnologies(ByVal Repository As EA.
Repository) As Object
 EA_OnInitializeTechnologies = My.Resources.MyTechnology
End Function
```

### 18.3.5.18.2 EA\_OnPreActivateTechnology

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also                                                                                                                                                                                     |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><b>EA_OnPreActivateTechnology</b> notifies Add-Ins that an MDG Technology resource is about to be activated in the model. This event occurs when a user selects to activate an MDG Technology resource in the model (by clicking on the <b>Set Active</b> button on the <b>MDG Technologies</b> dialog or by selecting the technology in the list box in the <b>Default Tools</b> toolbar).</p> <p>The notification is provided immediately after the user attempts to activate the MDG Technology, so that the Add-In can permit or disable activation of the Technology.</p> <p>Also look at <b>EA_OnPostActivateTechnology</b>.</p> | <p><a href="#">MDG Technologies</a> <sup>[1033]</sup></p> <p><a href="#">Default Tools Toolbar</a> <sup>[107]</sup></p> <p><a href="#">EA_OnPostActivateTechnology</a> <sup>[2037]</sup></p> |

#### Syntax

**Function** EA\_OnPreActivateTechnology(*Repository As EA.Repository, Info As EA.EventProperties*) As Boolean

The *EA\_OnPreActivateTechnology* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                                                                               |
|------------|------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the MDG Technology to be activated: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID.</li> </ul> |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                 |

#### Return Value

Return **True** to enable activation of the MDG Technology resource in the model. Return **False** to disable activation of the MDG Technology resource.

### 18.3.5.18.3 EA\_OnPostActivateTechnology

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See Also                                                                                                                                                                                    |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><b>EA_OnPostActivateTechnology</b> notifies Add-Ins that an MDG Technology resource has been activated in the model. This event occurs when a user activates an MDG Technology resource in the model (by clicking on the <b>Set Active</b> button on the <b>MDG Technologies</b> dialog or by selecting the technology in the list box in the <b>Default Tools</b> toolbar). The notification is provided immediately after the user succeeds in activating the MDG Technology, so that the Add-In can update the Technology if necessary.</p> <p>Also look at <b>EA_OnPreActivateTechnology</b>.</p> | <p><a href="#">MDG Technologies</a> <sup>[1033]</sup></p> <p><a href="#">Default Tools Toolbar</a> <sup>[107]</sup></p> <p><a href="#">EA_OnPreActivateTechnology</a> <sup>[2036]</sup></p> |

#### Syntax

**Function** EA\_OnPostActivateTechnology(*Repository As EA.Repository, Info As EA.EventProperties*)

The *EA\_OnPostActivateTechnology* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                                                                               |
|------------|------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the MDG Technology to be activated: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID.</li> </ul> |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                 |

#### Return Value

Return **True** if the MDG Technology resource is updated during this notification. Return **False** otherwise.

### 18.3.5.18.4 EA\_OnPreDeleteTechnology

**Deprecated** - refers to deleting a technology through the **Resources** window; this process is no longer recommended. See:

| Topic                       | Link                                                          |
|-----------------------------|---------------------------------------------------------------|
| EA_OnInitializeTechnologies | <a href="#">EA_OnInitializeTechnologies</a> <sup>[2035]</sup> |
| EA_OnPreActivateTechnology  | <a href="#">EA_OnPreActivateTechnology</a> <sup>[2036]</sup>  |
| EA_OnPostActivateTechnology | <a href="#">EA_OnPostActivateTechnology</a> <sup>[2037]</sup> |

Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also                                                |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| Details | <p><b>EA_OnPreDeleteTechnology</b> notifies Add-Ins that an MDG Technology resource is about to be deleted from the model. This event occurs when a user deletes an MDG Technology resource from the model. The notification is provided immediately after the user confirms their request to delete the MDG Technology, so that the Add-In can disable deletion of the MDG Technology.</p> <p>Also look at <b>EA_OnDeleteTechnology</b>.</p> | <a href="#">EA_OnDeleteTechnology</a> <sup>[2038]</sup> |

Syntax

**Function** EA\_OnPreDeleteTechnology(*Repository* As EA.Repository, *Info* As EA.EventProperties) As Boolean

The *EA\_OnPreDeleteTechnology* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                                                                             |
|------------|------------------------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | Contains the following <i>EventProperty</i> objects for the MDG Technology to be deleted: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID.</li> </ul> |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                               |

Return Value

Return **True** to enable deletion of the MDG Technology resource from the model. Return **False** to disable deletion of the MDG Technology resource.

**18.3.5.18.5 EA\_OnDeleteTechnology**

**Deprecated** - refers to deleting a technology through the **Resources** window; this process is no longer recommended. See:

| Topic                       | Link                                                          |
|-----------------------------|---------------------------------------------------------------|
| EA_OnInitializeTechnologies | <a href="#">EA_OnInitializeTechnologies</a> <sup>[2035]</sup> |
| EA_OnPreActivateTechnology  | <a href="#">EA_OnPreActivateTechnology</a> <sup>[2036]</sup>  |
| EA_OnPostActivateTechnology | <a href="#">EA_OnPostActivateTechnology</a> <sup>[2037]</sup> |

**Topics:**

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                      | See Also                                                   |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
| Details | <p><b>EA_OnDeleteTechnology</b> notifies Add-Ins that an MDG Technology resource has been deleted from the model.</p> <p>This event occurs after a user has deleted an MDG Technology resource from the model. Add-Ins that require an MDG Technology resource to be loaded can catch this event to disable certain functionality.</p> <p>Also look at <b>EA_OnPreDeleteTechnology</b>.</p> | <a href="#">EA_OnPreDeleteTechnology</a> <sup>[2037]</sup> |

**Syntax****Sub EA\_OnDeleteTechnology(Repository As EA.Repository, Info As EA.EventProperties)**

The *EA\_OnDeleteTechnology* function syntax contains the following elements:

| Parameter  | Type                                                 | Direction | Description                                                                                                                                                               |
|------------|------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a> <sup>[1844]</sup> | IN        | Contains the following <i>EventProperty</i> objects:<br><br><i>TechnologyID</i> : A string value corresponding to the MDG Technology ID.                                  |
| Repository | <a href="#">EA.Repository</a> <sup>[1855]</sup>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

**Return Value**

None.

**18.3.5.18.6 EA\_OnImportTechnology**

**Deprecated** - refers to deleting a technology through the **Resources** window; this process is no longer recommended. See:

| Topic                       | Link                                                          |
|-----------------------------|---------------------------------------------------------------|
| EA_OnInitializeTechnologies | <a href="#">EA_OnInitializeTechnologies</a> <sup>[2035]</sup> |
| EA_OnPreActivateTechnology  | <a href="#">EA_OnPreActivateTechnology</a> <sup>[2036]</sup>  |
| EA_OnPostActivateTechnology | <a href="#">EA_OnPostActivateTechnology</a> <sup>[2037]</sup> |

**Topics:**

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                 | See Also |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>EA_OnImportTechnology</b> notifies Add-Ins that you have imported an MDG Technology resource into the model.</p> <p>This event occurs after you have imported an MDG Technology resource into the model. Add-Ins that require an MDG Technology resource to be loaded can catch this Add-In to enable certain functionality.</p> |          |

### Syntax

**Sub EA\_OnImportTechnology(Repository As EA.Repository, Info As EA.EventProperties)**

The *EA\_OnImportTechnology* function syntax contains the following elements:

| Parameter  | Type                                                        | Direction | Description                                                                                                                                                               |
|------------|-------------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Info       | <a href="#">EA.EventProperties</a><br><small>[1844]</small> | IN        | Contains the following <i>EventProperty</i> objects:<br><br><i>TechnologyID</i> : A string value corresponding to the MDG Technology ID.                                  |
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small>      | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

None.

## 18.3.6 Custom Views

### Topics:

| Topic         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | See Also                                                        |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| General Usage | <p>Enterprise Architect enables custom windows to be inserted as tabs in the <b>Diagram View</b> that appears at the center of the Enterprise Architect frame.</p> <p><b>Creating a custom view</b> enables you to easily and quickly tab between a custom interface and diagrams and other views normally provided by Enterprise Architect.</p> <p>Uses for this facility include:</p> <ul style="list-style-type: none"> <li>• Reports and graphs showing summary data of the model</li> <li>• Alternative views of a diagram</li> <li>• Alternative views of the model</li> <li>• Views of external data related to model data</li> <li>• Documentation tools</li> </ul> | <a href="#">Creating a custom view</a><br><small>[2041]</small> |

### 18.3.6.1 Create a Custom View

#### Topics:

| Topic                                     | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also                                                    |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| <p><b>How to create a Custom View</b></p> | <p>A custom view must be designed as an ActiveX custom control and inserted through the automation interface.</p> <p>ActiveX custom controls can be created using most well-known programming tools including Microsoft Visual Studio.NET. See the documentation provided by the relevant vendor on how to create a custom control to produce an OCX file.</p> <p>Once the custom control has been created and registered on the target system, it can be added through the <b>AddTab()</b> method of the <b>Repository</b> object.</p> <p>While it is possible to call <b>AddTab()</b> from any automation client, it is likely that you would call it from an Add-In, and that Add-In is defined in the same OCX that provides the custom view.</p> <p>Example C# code is shown below:</p> <pre> public class AddIn {     UserControl1 m_MyControl;      public void EA_Connect ( EA.Repository Rep)     {         public object EA_GetMenuItems( EA. Repository Repository, string Location, string MenuName)         {             if ( MenuName == "" )                 return "- &amp;C# Control Demo";             else             {                 String( ) ret = {"&amp;Create", "&amp;Show Button"};                 return ret;             }         }          public void EA_MenuClick( EA.Repository Rep, string Location, string MenuName, string ItemName)         {             if ( ItemName == "&amp;Create" )                 m_MyControl = (UserControl1) Rep.AddTab(" C# Demo", "Cont Demo. UserControl1");             else                 m_MyControl.ShowButton();         }     } } </pre> | <p><a href="#">Repository Class</a> <small>1855</small></p> |

### 18.3.7 Custom Docked Window

#### Topics:

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See Also                                           |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Abstract</b> | <p>Enterprise Architect enables a custom docked window to be add to Enterprise Architect and shown and docked in the same way as other Enterprise Architect docked windows.</p> <p>A custom docked window must be designed as an ActiveX custom control and inserted through the automation interface.</p> <p>ActiveX custom controls can be created using most well-known programming tools including Microsoft Visual Studio.NET. See the documentation provided by the relevant vendor on how to create a custom control to produce an OCX file.</p> <p>Once the custom control has been created and registered on the target system, it can be added through the <b>AddWindow()</b> method of the <b>Repository</b> object.</p> <p>While it is possible to call <b>AddWindow()</b> from any automation client, it is likely that you would call it from an Add-In, and that Add-In is defined in the same OCX that provides the custom view.</p> | <a href="#">Repository Class</a> <sup>[1855]</sup> |

### 18.3.8 MDG Add-Ins

#### Topics:

| Topic           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See Also                                                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Abstract</b> | <p>MDG Add-Ins are specialized types of Add-Ins that have additional features and extra requirements for Add-In authors who want to contribute to Enterprise Architect's goal of Model Driven Generation</p> <p>Unlike general Add-In events, MDG Add-In events are only sent to the Add-In that has taken ownership of an Enterprise Architect model branch on a particular PC</p> <p>One of the additional responsibilities of an MDG Add-In is to take ownership of a branch of an Enterprise Architect model, which is done through the <i>MDG_Connect</i> event</p> <p>MDG Add-Ins identify themselves as such during <i>EA_Connect</i> by returning the string <b>MDG</b></p> <p>Unlike ordinary Add-Ins, responding to <i>MDG Add-In</i> events is not optional, and methods must be published for each of the <i>MDG Events</i></p> <p>Two examples of MDG Add-Ins are the commercially available MDG Link for Eclipse and MDG Link for Visual Studio, published by <i>Sparx Systems</i></p> | <a href="#">MDG_Connect</a> <sup>[2044]</sup><br><a href="#">EA_Connect</a> <sup>[1977]</sup><br><a href="#">MDG Events</a> <sup>[2042]</sup><br><a href="#">Sparx Systems</a><br>(Online Resource) |

#### 18.3.8.1 MDG Events

An MDG Add-In must respond to all MDG Events. These events usually identify processes such as Build, Run, Synchronize, PreMerge and PostMerge, amongst others.

An MDG Link Add-In is expected to implement some form of forward and reverse engineering capability within Enterprise Architect, and as such requires access to a specific set of events, all to do with generation,

synchronization and general processes concerned with converting models to code and code to models.

| Topic                    | Link                                                       |
|--------------------------|------------------------------------------------------------|
| MDG_BuildProject         | <a href="#">MDG_BuildProject</a> <sup>[2043]</sup>         |
| MDG_Connect              | <a href="#">MDG_Connect</a> <sup>[2044]</sup>              |
| MDG_Disconnect           | <a href="#">MDG_Disconnect</a> <sup>[2045]</sup>           |
| MDG_GetConnectedPackages | <a href="#">MDG_GetConnectedPackages</a> <sup>[2046]</sup> |
| MDG_GetProperty          | <a href="#">MDG_GetProperty</a> <sup>[2046]</sup>          |
| MDG_Merge                | <a href="#">MDG_Merge</a> <sup>[2047]</sup>                |
| MDG_NewClass             | <a href="#">MDG_NewClass</a> <sup>[2049]</sup>             |
| MDG_PostGenerate         | <a href="#">MDG_PostGenerate</a> <sup>[2050]</sup>         |
| MDG_PostMerge            | <a href="#">MDG_PostMerge</a> <sup>[2051]</sup>            |
| MDG_PreGenerate          | <a href="#">MDG_PreGenerate</a> <sup>[2051]</sup>          |
| MDG_PreMerge             | <a href="#">MDG_PreMerge</a> <sup>[2052]</sup>             |
| MDG_PreReverse           | <a href="#">MDG_PreReverse</a> <sup>[2053]</sup>           |
| MDG_RunExe               | <a href="#">MDG_RunExe</a> <sup>[2054]</sup>               |
| MDG_View                 | <a href="#">MDG_View</a> <sup>[2054]</sup>                 |

### 18.3.8.1.1 MDGBuild Project

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                       | See Also                                     |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| Details | <p><b>MDG_BuildProject</b> enables the Add-In to handle file changes caused by generation. This function is called in response to a user selecting the <b>Add-Ins   Build Project</b> menu option.</p> <p>Respond to this event by compiling the project source files into a running application.</p> <p>Also look at <b>MDG_RunExe</b>.</p> | <a href="#">MDG_RunExe</a> <sup>[2054]</sup> |

#### Syntax

#### Sub MDG\_BuildProject(Repository As EA.Repository, PackageGuid As String)

The *MDG\_BuildProject* function syntax contains the following elements:

| Parameter   | Type   | Direction | Description                                                |
|-------------|--------|-----------|------------------------------------------------------------|
| PackageGuid | String | IN        | The GUID identifying the Enterprise Architect package sub- |

| Parameter  | Type                                    | Direction | Description                                                                                                                                                               |
|------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            |                                         |           | tree that is controlled by the Add-In.                                                                                                                                    |
| Repository | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

None.

#### 18.3.8.1.2 MDGConnect

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | See Also                                         |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Details | <p><b>MDG_Connect</b> enables the Add-In to handle user driven request to connect a model branch to an external application. This function is called when the user attempts to connect a particular Enterprise Architect package to an as yet unspecified external project. This event enables the Add-In to interact with the user to specify such a project.</p> <p>The Add-In is responsible for retaining the connection details, which should be stored on a per-user or per-workstation basis. That is, users who share a common Enterprise Architect model over a network should be able to connect and disconnect to external projects independently of one another.</p> <p>The Add-In should therefore not store connection details in an Enterprise Architect repository. A suitable place to store such details would be:</p> <pre>SHGet Folder Path( . . CSI_DL_APPDATA. . ) \ Addi nName.</pre> <p>The <i>PackageGuid</i> parameter is the same identifier as required for most events relating to the MDG Add-In. Therefore it is recommended that the connection details be indexed using the PackageGuid value.</p> <p>The <i>PackageID</i> parameter is provided to aid fast retrieval of package details from Enterprise Architect, should this be required.</p> <p>Also look at <b>MDG_Disconnect</b>.</p> | <p><a href="#">MDG_Disconnect</a><br/>[2045]</p> |

#### Syntax

**Function** MDG\_Connect(*Repository As EA.Repository, PackageID as Long, PackageGuid As String*) *As Long*

The *MDG\_Connect* function syntax contains the following elements:

| Parameter   | Type                                                   | Direction | Description                                                                                                                                                               |
|-------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PackageGuid | String                                                 | IN        | The unique ID identifying the project provided by the Add-In when a connection to a project branch of an Enterprise Architect model was first established.                |
| PackageID   | Long                                                   | IN        | The <i>PackageID</i> of the Enterprise Architect package the user has requested to have connected to an external project.                                                 |
| Repository  | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

Returns a non-zero to indicate that a connection has been made; a zero indicates that the user has not nominated a project and connection should not proceed.

### 18.3.8.1.3 MDGDisconnect

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                              | See Also                                      |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| Details | <b>MDG_Disconnect</b> enables the Add-In to respond to user requests to disconnect the model branch from an external project. This function is called when the user attempts to disconnect an associated external project. The Add-In is required to delete the details of the connection.<br><br>Also look at <b>MDG_Connect</b> . | <a href="#">MDG_Connect</a> <sup>[2044]</sup> |

### Syntax

**Function** MDG\_Disconnect(*Repository As EA.Repository, PackageGuid As String*) **As Long**

The *MDG\_Disconnect* function syntax contains the following elements:

| Parameter   | Type                                                   | Direction | Description                                                                                                                                                               |
|-------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PackageGuid | String                                                 | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                          |
| Repository  | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

Returns a non-zero to indicate that a disconnection has occurred enabling Enterprise Architect to update the user interface. A zero indicates that the user has not disconnected from an external project.

### 18.3.8.1.4 MDGGetConnectedPackages

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                          | See Also                                          |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| Details | <p><b>MDG_GetConnectedPackages</b> enables the Add-In to return a list of current connection between Enterprise Architect and an external application. This function is called when the Add-In is first loaded, and is expected to return a list of the available connections to external projects for this Add-In.</p> <p>Also look at <b>MDG_Connect</b>.</p> | <a href="#">MDG_Connect</a> <small>[2044]</small> |

#### Syntax

#### Function MDG\_GetConnectedPackages(*Repository As EA.Repository*) As Variant

The *MDG\_GetConnectedPackages* function syntax contains the following elements:

| Parameter  | Type                                                | Direction | Description                                                                                                                                                               |
|------------|-----------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a> <small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

Returns an array of GUID strings representing individual Enterprise Architect packages.

### 18.3.8.1.5 MDGGetProperty

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | See Also |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <p><b>MDG_GetProperty</b> provides miscellaneous Add-In details to Enterprise Architect. This function is called by Enterprise Architect to poll the Add-In for information relating to the <i>PropertyName</i>. This event should occur in as short a duration as possible as Enterprise Architect does not cache the information provided by the function.</p> <p>Values corresponding to the following <i>PropertyNames</i> must be provided:</p> <ul style="list-style-type: none"> <li>• <b>IconID</b> - Return the name of a DLL and a resource identifier in the format <i>#ResID</i>, where the resource ID indicates an Icon; for example, <i>c:\program files\myapp\myapp.dll#101</i></li> <li>• <b>Language</b> - Return the default language that Classes should be assigned when they are created in Enterprise Architect</li> <li>• <b>HiddenMenus</b> - Return one or more values from the</li> </ul> |          |

| Topic | Detail                                                                                                                                                                         | See Also |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|       | <p><i>MDGMenu</i> enumeration to hide menus that do not apply to your Add-In. For example:</p> <pre>if ( PropertyName == " HiddenMenus" ) return mgBuildProject + mgRun;</pre> |          |

### Syntax

**Function** *MDG\_GetProperty(Repository As EA.Repository, PackageGuid As String, PropertyName As String) As Variant*

The *MDG\_GetProperty* function syntax contains the following elements:

| Parameter           | Type                                    | Direction | Description                                                                                                                                                               |
|---------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PackageGuid</b>  | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                          |
| <b>PropertyName</b> | <i>String</i>                           | IN        | The name of the property that is used by Enterprise Architect. See <i>Details</i> for the possible values.                                                                |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently-open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

See *Details*, above.

## 18.3.8.1.6 MDGMerge

### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See Also                                                                                                                                 |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p><b>MDG_Merge</b> enables the Add-In to jointly handle changes to both the model branch and the code project that the model branch is connected to. This event should be called whenever the user has asked to merge their model branch with its connected code project, or whenever the user has established a new connection to a code project. The purpose of this event is to enable the Add-In to interact with the user to perform a merge between the model branch and the connected project.</p> | <p><a href="#">MDG_Connect</a> [2044]</p> <p><a href="#">MDG_PreMerge</a><br/>[2052]</p> <p><a href="#">MDG_PostMerge</a><br/>[2051]</p> |

### Syntax

**Function** *MDG\_Merge(Repository As EA.Repository, PackageGuid As String, SynchObjects As Variant, SynchType As String, ExportObjects As Variant, ExportFiles As Variant, ImportFiles As Variant, IgnoreLocked As String, Language As String) As Long*

The *MDG\_Merge* function syntax contains the following elements:

| Parameter            | Type                                    | Direction | Description                                                                                                                                                                                                                                                                                                              |
|----------------------|-----------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ExportFiles</b>   | <i>Variant</i>                          | OUT       | A string array containing the list of files for each model object chosen for export by the Add-In. Each entry in this array must have a corresponding entry in the <i>ExportObjects</i> parameter at the same array index, so <i>ExportFiles(2)</i> must contain the filename of the object by <i>ExportObjects(2)</i> . |
| <b>ExportObjects</b> | <i>Variant</i>                          | OUT       | The string array containing the list of new model objects (in <i>Object ID</i> format) to be exported by Enterprise Architect to the code project.                                                                                                                                                                       |
| <b>IgnoreLocked</b>  | <i>String</i>                           | OUT       | A value indicating whether to ignore any files locked by the code project (that is, "TRUE" or "FALSE").                                                                                                                                                                                                                  |
| <b>ImportFiles</b>   | <i>Variant</i>                          | OUT       | A string array containing the list of code files made available to the code project to be newly imported to the model. Enterprise Architect imports each file listed in this array for import into the connected model branch.                                                                                           |
| <b>Language</b>      | <i>String</i>                           | OUT       | The string value containing the name of the code language supported by the code project connected to the model branch.                                                                                                                                                                                                   |
| <b>PackageGuid</b>   | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                                                                                                                                                         |
| <b>Repository</b>    | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                |
| <b>SynchObjects</b>  | <i>Variant</i>                          | OUT       | A string array containing a list of objects ( <i>Object ID</i> format) to be jointly synchronized between the model branch and the project. See <a href="#">below</a> [2049] for the format of the Object IDs.                                                                                                           |
| <b>SynchType</b>     | <i>String</i>                           | OUT       | The value determining the user-selected type of synchronization to take place. See <a href="#">below</a> [2048] for a list of valid values.                                                                                                                                                                              |

### Return Value

Return a non-zero if the merge operation completed successfully and a zero value when the operation has been unsuccessful.

### Merge

A merge consists of three major operations:

- **Export:** Where newly created model objects are exported into code and made available to the code project.
- **Import:** Where newly created code objects, Classes and such things are imported into the model.
- **Synchronize:** Where objects available both to the model and in code are jointly updated to reflect changes made in either the model, code project or both

### Synchronize Type

The *Synchronize* operation can take place in one of four different ways. Each of these ways corresponds to a value returned by *SynchType*:

- None: (*SynchType* = 0) No synchronization is to be performed
- Forward: (*SynchType* = 1) Forward synchronization, between the model branch and the code project is

- to occur
- Reverse: (*SynchType* = 2) Reverse synchronization, between the code project and the model branch is to occur
  - Both: (*SynchType* = 3) Reverse, then Forward synchronization's are to occur

### Object ID Format

Each of the Object IDs listed in the string arrays described above should be composed in the following format:

```
(@namespace) * (#class) * ($attribute | %operation | : property) *
```

#### 18.3.8.1.7 MDGNewClass

### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                            | See Also                                  |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Details | <p><b>MDG_NewClass</b> enables the Add-In to alter details of a Class before it is created.</p> <p>This method is called when Enterprise Architect generates a new Class, and requires information relating to assigning the language and file path. The file path should be passed back as a return value and the language should be passed back via the language parameter.</p> | <a href="#">MDG_PreGenerate</a><br>[2057] |

### Syntax

**Function** *MDG\_NewClass(Repository As EA.Repository, PackageGuid As String, CodeID As String, Language As String) As String*

The *MDG\_NewClass* function syntax contains the following elements:

| Parameter          | Type                                    | Direction | Description                                                                                                                                                               |
|--------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CodeID</b>      | <i>String</i>                           | IN        | A string used to identify the code element before it is created, for more information see <a href="#">MDG_View</a> [2057].                                                |
| <b>Language</b>    | <i>String</i>                           | OUT       | A string used to identify the programming language for the new Class. The language must be supported by Enterprise Architect.                                             |
| <b>PackageGuid</b> | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                          |
| <b>Repository</b>  | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

### Return Value

Returns a string containing the file path that should be assigned to the Class.

### 18.3.8.1.8 MDGPostGenerate

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See Also                                                 |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| Details | <p><b>MDG_PostGenerate</b> enables the Add-In to handle file changes caused by generation.</p> <p>This event is called after Enterprise Architect has prepared text to replace the existing contents of a file. Responding to this event enables the Add-In to write to the linked application's user interface rather than modify the file directly.</p> <p>When the contents of a file are changed, Enterprise Architect passes <i>FileContents</i> as a non-empty string. New files created as a result of code generation are also sent through this mechanism, enabling Add-Ins to add new files to the linked project's file list.</p> <p>When new files are created Enterprise Architect passes <i>FileContents</i> as an empty string. When a non-zero is returned by this function, the Add-In has successfully written the contents of the file. A zero value for the return indicates to Enterprise Architect that the file must be saved.</p> | <a href="#">MDG_PreGenerate</a><br><small>[2051]</small> |

#### Syntax

**Function** `MDG_PostGenerate(Repository As EA.Repository, PackageGuid As String, FilePath As String, FileContents As String) As Long`

The `MDG_PostGenerate` function syntax contains the following elements:

| Parameter           | Type                                                   | Direction | Description                                                                                                                                                               |
|---------------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FileContents</b> | <i>String</i>                                          | IN        | A string containing the proposed contents of the file.                                                                                                                    |
| <b>FilePath</b>     | <i>String</i>                                          | IN        | The path of the file Enterprise Architect intends to overwrite.                                                                                                           |
| <b>PackageGuid</b>  | <i>String</i>                                          | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                          |
| <b>Repository</b>   | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

#### Return Value

Return value depends on the type of event that this function is responding to (see **Details**, above). This function is required to handle two separate and distinct cases.

### 18.3.8.1.9 MDGPostMerge

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                | See Also                                                                                                                                                                                                                                     |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p>MDG_PostMerge is called after a merge process has been completed.</p> <p>This function is called by Enterprise Architect after the merge process has been completed.</p> <p>File save checking should not be performed with this function, but should be handled by <b>MDG_PreGenerate</b>, <b>MDG_PostGenerate</b> and <b>MDG_PreReverse</b>.</p> | <p><a href="#">MDG_PreGenerate</a><br/>[2051]</p> <p><a href="#">MDG_PostGenerate</a><br/>[2050]</p> <p><a href="#">MDG_PreReverse</a><br/>[2053]</p> <p><a href="#">MDG_PreMerge</a><br/>[2052]</p> <p><a href="#">MDG_Merge</a> [2047]</p> |

#### Syntax

*Function MDG\_PostMerge(Repository As EA.Repository, PackageGuid As String) As Long*

The *MDG\_PostMerge* function syntax contains the following elements:

| Parameter   | Type                                    | Direction | Description                                                                                                                                                               |
|-------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository  | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| PackageGuid | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                          |

#### Return Value

Return a zero value if the post-merge process has failed, a non-zero return indicates that the post-merge has been successful. Enterprise Architect assumes a non-zero return if this method is not implemented

### 18.3.8.1.10 MDGPreGenerate

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                              | See Also                                           |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| Details | <p><b>MDG_PreGenerate</b> enables the Add-In to deal with unsaved changes. This function is called immediately before Enterprise Architect attempts to generate files from the model. A possible use of this function would be to prompt the user to save unsaved source files.</p> | <p><a href="#">MDG_PostGenerate</a><br/>[2050]</p> |

#### Syntax

**Function MDG\_PreGenerate(Repository As EA.Repository, PackageGuid As String) As Long**

The *MDG\_PreGenerate* function syntax contains the following elements:

| Parameter   | Type                                                   | Direction | Description                                                                                                                                                               |
|-------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository  | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| PackageGuid | String                                                 | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                          |

**Return Value**

Return a zero value to abort generation. Any other value enables the generation to continue.

**18.3.8.1.11 MDGPreMerge****Topics:**

| Topic   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See Also                                                                                                                                                                                                                                                                                                                     |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Details | <p><b>MDG_PreMerge</b> is called after a merge process has been initiated by the user and before Enterprise Architect performs the merge process.</p> <p>This event is called after a user has performed their interactions with the merge screen and has confirmed the merge with the OK button, but before Enterprise Architect performs the merge process using the data provided by the <i>MDG_Merge</i> call, before any changes have been made to the model or the connected project.</p> <p>This event is made available to provide the Add-In with the opportunity to generally set internal Add-In flags to augment the <i>MDG_PreGenerate</i>, <i>MDG_PostGenerate</i> and <i>MDG_PreReverse</i> events.</p> <p>File save checking should not be performed with this function, but should be handled by <b>MDG_PreGenerate</b>, <b>MDG_PostGenerate</b> and <b>MDG_PreReverse</b>.</p> <p>Also look at <b>MDG_Merge</b> and <b>MDG_PostMerge</b>.</p> | <p><a href="#">MDG_PreGenerate</a><br/><small>[2051]</small></p> <p><a href="#">MDG_PostGenerate</a><br/><small>[2050]</small></p> <p><a href="#">MDG_PreReverse</a><br/><small>[2053]</small></p> <p><a href="#">MDG_Merge</a><br/><small>[2047]</small></p> <p><a href="#">MDG_PostMerge</a><br/><small>[2051]</small></p> |

**Syntax****Function MDG\_PreMerge(Repository As EA.Repository, PackageGuid As String) As Long**

The *MDG\_PreMerge* function syntax contains the following elements:

| Parameter  | Type                                                   | Direction | Description                                                                                                                                                               |
|------------|--------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository | <a href="#">EA.Repository</a><br><small>[1855]</small> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |

| Parameter          | Type          | Direction | Description                                                                                      |
|--------------------|---------------|-----------|--------------------------------------------------------------------------------------------------|
| <b>PackageGuid</b> | <i>String</i> | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In. |

### Return Value

A return value of zero indicates that the merge process can not occur. If the value is not zero the merge process proceeds. If this method is not implemented then it is assumed that a merge process is used.

### 18.3.8.1.12 MDGPreReverse

#### Topics:

| Topic          | Detail                                                                                                                                                                                                                                                                                                                                                             | See Also                                                                                             |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| <b>Details</b> | <p>MDG_PreReverse enables the Add-In to save file changes before being imported into Enterprise Architect.</p> <p>This function operates on a list of files that are about to be reverse-engineered into Enterprise Architect. If the user is working on unsaved versions of these files in an editor, you could either prompt the user or save automatically.</p> | <p><a href="#">MDG_PostGenerate</a><br/>[2050]</p> <p><a href="#">MDG_PreGenerate</a><br/>[2051]</p> |

### Syntax

**Sub MDG\_PreReverse(Repository As EA.Repository, PackageGuid As String, FilePaths As Variant)**

The *MDG\_PreReverse* function syntax contains the following elements:

| Parameter          | Type                                    | Direction | Description                                                                                                                                                               |
|--------------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b>  | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| <b>PackageGuid</b> | <i>String</i>                           | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                          |
| <b>FilePaths</b>   | <i>String array</i>                     | IN        | An array of filepaths pointed to the files that are to be reverse engineered.                                                                                             |

### Return Value

None.

### 18.3.8.1.13 MDGRunExe

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                   | See Also                                   |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| Details | <b>MDG_RunExe</b> enables the Add-In to run the target application. This function is called when the user selects the <b>Add-Ins   Run Exe</b> menu option. Respond to this event by launching the compiled application. | <a href="#">MDG_BuildProject</a><br>[2043] |

#### Syntax

**Sub MDG\_RunExe(Repository As EA.Repository, PackageGuid As String)**

The *MDG\_RunExe* function syntax contains the following elements:

| Parameter   | Type                                    | Direction | Description                                                                                                                                                               |
|-------------|-----------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Repository  | <a href="#">EA.Repository</a><br>[1855] | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information. |
| PackageGuid | String                                  | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                          |

#### Return Value

None.

### 18.3.8.1.14 MDGView

#### Topics:

| Topic   | Detail                                                                                                                                                                                                                                                                      | See Also |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| Details | <b>MDG_View</b> enables the Add-In to display user specified code elements. This function is called by Enterprise Architect when the user asks to view a particular code element. This enables the Add-In to present that element in its own way, usually in a code editor. |          |

#### Syntax

**Function MDG\_View(Repository As EA.Repository, PackageGuid As String, CodeID as String) As Long**

The *MDG\_View* function syntax contains the following elements:

| Parameter          | Type                 | Direction | Description                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------|----------------------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository</b>  | <i>EA.Repository</i> | IN        | An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.                                                                                                                                                                                                                                                                             |
| <b>PackageGuid</b> | <i>String</i>        | IN        | The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.                                                                                                                                                                                                                                                                                                                                                      |
| <b>CodeID</b>      | <i>String</i>        | IN        | Identifies the code element in the following format:<br><type>ElementPart<type>ElementPart..<br>where each element is preceded with a token identifying its type:<br>@ -namespace<br># - Class<br>\$ - attribute<br>% - operation<br>For example if a user has selected the <i>m_Name</i> attribute of <i>Class1</i> located in <i>namespace Name1</i> , the <i>class ID</i> would be passed through in the following format:<br>@Name1#Class1%m_Name |

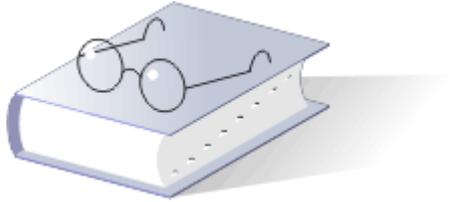
### Return Value

Return a non-zero value to indicate that the Add-In has processed the request. Returning a zero value results in Enterprise Architect employing the standard viewing process which is to launch the associated source file.

**Part**



## 19 Glossary



This topic provides a detailed glossary for Enterprise Architect.

|                             |                             |                             |                             |                             |                             |                             |                             |                             |                             |                             |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| <a href="#">A</a><br>[2058] | <a href="#">B</a><br>[2060] | <a href="#">C</a><br>[2061] | <a href="#">D</a><br>[2064] | <a href="#">E</a><br>[2066] | <a href="#">F</a><br>[2068] | <a href="#">G</a><br>[2069] | <a href="#">H</a><br>[2070] | <a href="#">I</a><br>[2071] | <a href="#">J</a><br>[2073] | <a href="#">L</a><br>[2074] |
| <a href="#">M</a><br>[2075] | <a href="#">N</a><br>[2077] | <a href="#">O</a><br>[2078] | <a href="#">P</a><br>[2079] | <a href="#">Q</a><br>[2082] | <a href="#">R</a><br>[2083] | <a href="#">S</a><br>[2085] | <a href="#">T</a><br>[2088] | <a href="#">U</a><br>[2090] | <a href="#">V</a><br>[2092] |                             |

## 19.1 A

| Name                       | Detail                                                                                                                                                                                                                                                                            | See also                     |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| <b>Abstract Class</b>      | A Class that cannot be directly instantiated.                                                                                                                                                                                                                                     | Concrete Class               |
| <b>Abstraction</b>         | The essential characteristics of an entity that distinguish it from all other kinds of entities. An abstraction defines a boundary relative to the perspective of the viewer.                                                                                                     |                              |
| <b>Action</b>              | The specification of an executable statement that forms an abstraction of a computation procedure. An action typically results in a change in the state of the system, and can be realized by sending a message to an object or modifying a connector or a value of an attribute. |                              |
| <b>Action Sequence</b>     | An expression that resolves to a sequence of actions.                                                                                                                                                                                                                             |                              |
| <b>Action State</b>        | A state that represents the execution of an atomic action, typically the invocation of an operation.                                                                                                                                                                              |                              |
| <b>Activation</b>          | The execution of an action.                                                                                                                                                                                                                                                       |                              |
| <b>Active Class</b>        | A Class whose instances are active objects. When instantiated, an active Class controls its execution. Rather than being invoked or activated by other objects, it can operate standalone, and define its own thread of behavior.                                                 | Active Object                |
| <b>Active Object</b>       | An object that owns a thread and can initiate control activity. An instance of active Class.                                                                                                                                                                                      | Active Class, Thread         |
| <b>Activity</b>            | Defines the bounds for the structural organization that contains a set of basic or fundamental behaviors. It can be used to model procedural type application development for system design through to modeling business processes in organizational structures and work flow.    |                              |
| <b>Activity Diagram</b>    | A diagram used to model procedural type application development for system design through to modeling business processes in organizational structures and work flow.                                                                                                              |                              |
| <b>Activity Graph</b>      | A special case of a State Machine that is used to model processes involving one or more classifiers.                                                                                                                                                                              | State Chart Diagram          |
| <b>Actor ( Class )</b>     | A coherent set of roles that users of Use Cases play when interacting with these Use Cases. An Actor has one role for each Use Case with which it communicates.                                                                                                                   |                              |
| <b>Actual Parameter</b>    | A binding for a parameter that resolves to a run-time instance.                                                                                                                                                                                                                   | Argument, (Formal) Parameter |
| <b>Aggregate ( Class )</b> | A Class that represents the 'whole' in an Aggregation (whole-part) relationship.                                                                                                                                                                                                  | Aggregation                  |
| <b>Aggregation</b>         | A special form of Association that specifies a whole-part relationship between the Aggregate (whole) and a component part.                                                                                                                                                        | Composition                  |
| <b>Analysis</b>            | The part of the software development process whose primary purpose is to formulate a model of the problem domain. Analysis focuses on what to do, design focuses                                                                                                                  | Design                       |

| Name                     | Detail                                                                                                                                                                                                                                                                                                                         | See also                                                 |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
|                          | on how to do it.                                                                                                                                                                                                                                                                                                               |                                                          |
| <b>Analysis Diagram</b>  | A diagram used to capture high level business processes and early models of system behavior and elements. It is less formal than some other diagrams, but provides a good means of capturing the essential business characteristics and requirements.                                                                          |                                                          |
| <b>Analysis Time</b>     | Refers to something that occurs during an analysis phase of the software development process.                                                                                                                                                                                                                                  | Modeling Time,<br>Run Time, Compile Time,<br>Design Time |
| <b>Architecture</b>      | The organizational structure and associated behavior of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include Classes, Components and subsystems.        |                                                          |
| <b>Argument</b>          | A binding for a parameter that resolves to a run-time instance.                                                                                                                                                                                                                                                                | Actual Parameter<br>(Formal) Parameter                   |
| <b>Artifact</b>          | A physical piece of information that is used or produced by a business or development process. Examples of Artifacts include models, source files, scripts, and binary executable files. An Artifact can constitute the implementation of a deployable component.                                                              | Product,<br>Component                                    |
| <b>Assembly</b>          | A connector that bridges the required interface of a component with the provided interface of a second component.                                                                                                                                                                                                              |                                                          |
| <b>Association</b>       | The semantic relationship between two or more classifiers that specifies connections among their instances.                                                                                                                                                                                                                    | Link                                                     |
| <b>Association Class</b> | A model element that has both Association and Class properties. An Association Class can be seen as an Association that also has Class properties, or as a Class that also has Association properties.                                                                                                                         | Class                                                    |
| <b>Association End</b>   | The endpoint of an Association, which connects the Association to a classifier.                                                                                                                                                                                                                                                | Classifier,<br>Link End                                  |
| <b>Attribute</b>         | A feature within a classifier that describes a range of values that instances of the classifier can hold.                                                                                                                                                                                                                      |                                                          |
| <b>Auxiliary Class</b>   | A stereotyped Class that supports another more central or fundamental Class, typically by implementing secondary logic or control flow. Auxiliary Classes are typically used together with focus Classes, and are particularly useful for specifying the secondary business logic or control flow of components during design. | Focus Class                                              |

## 19.2 B

| Name                           | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also          |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>Behavior</b>                | The observable effects of an operation or event, including its results.                                                                                                                                                                                                                                                                                                                                                                                                                                    |                   |
| <b>Behavioral Diagram</b>      | A diagram that depicts the behavioral features of a system or business process. Behavioral diagrams include Activity diagrams, State Machine diagrams, Communication diagrams, Interaction Overview diagrams, Sequence diagrams, Timing diagrams and Use Case diagrams.                                                                                                                                                                                                                                    |                   |
| <b>Behavioral Feature</b>      | A dynamic feature of a model element, such as an operation or method.                                                                                                                                                                                                                                                                                                                                                                                                                                      |                   |
| <b>Behavioral Model aspect</b> | A model aspect that emphasizes the behavior of the instances in a system, including their methods, collaborations, and state histories.                                                                                                                                                                                                                                                                                                                                                                    |                   |
| <b>Binary Association</b>      | An Association between two Classes. A special case of an N-ary Association.                                                                                                                                                                                                                                                                                                                                                                                                                                | N-ary Association |
| <b>Binding</b>                 | The creation of a model element from a template by supplying arguments for the parameters of the template.                                                                                                                                                                                                                                                                                                                                                                                                 |                   |
| <b>Bookmark</b>                | A marker in a Rich Text Format document that enables you to link inner sections of a document into a master document (using the Word 'Insert File' function).                                                                                                                                                                                                                                                                                                                                              |                   |
| <b>Boolean</b>                 | An enumeration whose values are true and false.                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                   |
| <b>Boolean Expression</b>      | An expression that evaluates to a boolean value.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                   |
| <b>Boundary</b>                | <ol style="list-style-type: none"> <li>1. A stereotyped Class that models some system boundary – typically a user interface screen. It is used in the conceptual phase to capture user interaction with the system at a screen level (or some other boundary interface type). It is often used in Sequence and Robustness (Analysis) diagrams. It is the View in the Model-View-Controller pattern.</li> <li>2. A System Boundary element is used to delineate a particular part of the system.</li> </ol> |                   |

## 19.3 C

| Name                         | Detail                                                                                                                                                                                                                                                                                                               | See also                                                                |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| <b>C++</b>                   | An object-oriented programming language based on the earlier 'C' language.                                                                                                                                                                                                                                           |                                                                         |
| <b>Call</b>                  | An action state that invokes an operation on a classifier.                                                                                                                                                                                                                                                           |                                                                         |
| <b>Cardinality</b>           | The number of elements in a set.                                                                                                                                                                                                                                                                                     | Multiplicity                                                            |
| <b>CASE</b>                  | Computer Aided Software Engineering. A tool designed for the purpose of modeling and building software systems.                                                                                                                                                                                                      |                                                                         |
| <b>Child</b>                 | In a Generalization relationship, the specialization of another element, the parent.                                                                                                                                                                                                                                 | Subclass,Subtype,Parent                                                 |
| <b>Choice</b>                | A pseudo-state used to compose complex transitional paths, where the outgoing transition path is decided by dynamic, run-time conditions determined by the actions performed by the State Machine on the path leading to the choice.                                                                                 |                                                                         |
| <b>Class</b>                 | A description of a set of objects that share the same attributes, operations, methods, relationships and semantics. A Class can use a set of interfaces to specify collections of operations it provides to its environment.                                                                                         | Interface, Object                                                       |
| <b>Class Diagram</b>         | A diagram that shows a collection of declarative (static) model elements, such as Classes, types, and their contents and relationships.                                                                                                                                                                              | Object Diagram                                                          |
| <b>Classification</b>        | The assignment of an object to a classifier.                                                                                                                                                                                                                                                                         | Dynamic Classification, Multiple Classification, Static Classification. |
| <b>Classifier</b>            | A mechanism that describes behavioral and structural features. Classifiers include Interfaces, Classes, datatypes, and components.                                                                                                                                                                                   |                                                                         |
| <b>Client</b>                | A classifier that requests a service from another classifier.                                                                                                                                                                                                                                                        | Supplier                                                                |
| <b>Collaboration</b>         | The specification of how an operation or classifier, such as a Use Case, is realized by a set of classifiers and Associations playing specific roles used in a specific way. The Collaboration defines an interaction.                                                                                               | Interaction                                                             |
| <b>Collaboration Diagram</b> | Used pre-UML 2.0. Now called a Communication diagram.                                                                                                                                                                                                                                                                |                                                                         |
| <b>Collaboration Use</b>     | Uses an occurrence to apply a pattern defined by a Collaboration to a specific situation.                                                                                                                                                                                                                            |                                                                         |
| <b>Combined Fragment</b>     | A combined fragment reflects a piece or pieces of interaction (called interaction operands) controlled by an interaction operator, whose corresponding boolean conditions are known as interaction constraints. It appears graphically as a transparent window, divided by horizontal dashed lines for each operand. |                                                                         |
| <b>Comment</b>               | An annotation attached to an element or a collection of                                                                                                                                                                                                                                                              | Constraint                                                              |

| Name                               | Detail                                                                                                                                                                                                                                                                                                                                             | See also                                            |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
|                                    | elements. A comment, or note, has no semantics.                                                                                                                                                                                                                                                                                                    |                                                     |
| <b>Communication Diagram</b>       | A diagram that shows the interactions between elements at run-time in much the same manner as a Sequence diagram. However, Communication diagrams are used to visualize inter-object relationships, while Sequence diagrams are more effective at visualizing processing over time.                                                                | Collaboration Diagram, Object Diagram               |
| <b>Compile Time</b>                | Refers to something that occurs during the compilation of a software module.                                                                                                                                                                                                                                                                       | Modeling Time, Run Time, Analysis Time, Design Time |
| <b>Component</b>                   | A modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces. A Component is typically specified by one or more classifiers (such as implementation Classes) that reside on it, and can be implemented by one or more artifacts (such as binary, executable, or script files).         | Module Artifact, Product                            |
| <b>Component Diagram</b>           | A diagram that shows the organizations and dependencies among Components.                                                                                                                                                                                                                                                                          |                                                     |
| <b>Composite ( Class )</b>         | A Class that is related to one or more Classes by a Composition relationship.                                                                                                                                                                                                                                                                      | Composition                                         |
| <b>Composite State</b>             | A State that consists of either concurrent (orthogonal) substates or sequential (disjoint) substates.                                                                                                                                                                                                                                              | Substate, Concurrent Substate, Disjoint Substate    |
| <b>Composite Structure Diagram</b> | A diagram that reflects the internal collaboration of Classes, Interfaces, or Components to describe a functionality. Composite Structure diagrams are similar to Class diagrams, except that they model a specific usage of the structure.                                                                                                        |                                                     |
| <b>Composition</b>                 | A form of Aggregation that requires that a part instance be included in at most one Composite at a time, and that the Composite object is responsible for the creation and destruction of the parts. Composition can be recursive.                                                                                                                 | Composite Aggregation                               |
| <b>Concrete Class</b>              | A Class that can be directly instantiated.                                                                                                                                                                                                                                                                                                         | Abstract Class                                      |
| <b>Concurrency</b>                 | The occurrence of two or more activities during the same time interval. Concurrency can be achieved by interleaving or simultaneously executing two or more threads.                                                                                                                                                                               | Thread                                              |
| <b>Concurrent Substate</b>         | A substate that can be held simultaneously with other substates contained in the same composite State.                                                                                                                                                                                                                                             | Composite State<br>Disjoint Substate                |
| <b>Connector</b>                   | A logical link between model elements. Can be structural, dynamic or possessive.                                                                                                                                                                                                                                                                   |                                                     |
| <b>Constraint</b>                  | <ol style="list-style-type: none"> <li>1. A semantic condition or restriction. Certain constraints are predefined in the UML, others can be user defined. Constraints are one of three extensibility mechanisms in UML.</li> <li>2. A rule or condition that applies to some element. It is often modeled as a pre- or post- condition.</li> </ol> | Tagged Value, Stereotype<br>Comment                 |

| Name                         | Detail                                                                                                                                                                                                                    | See also |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Container</b>             | 1. An instance that exists to contain other instances, and that provides operations to access or iterate over its contents.(for example, arrays, lists, sets).<br>2. A component that exists to contain other components. |          |
| <b>Containment Hierarchy</b> | A namespace hierarchy consisting of model elements, and the containment relationships that exist between them. A containment hierarchy forms a graph.                                                                     |          |
| <b>Context</b>               | A view of a set of related modeling elements for a particular purpose, such as specifying an operation.                                                                                                                   |          |
| <b>Continuation</b>          | A Continuation is used in seq and alt combined fragments, to indicate the branches of continuation an operand follows.                                                                                                    |          |
| <b>Control</b>               | A stereotyped Class that represents a controlling entity or manager. A Control organizes and schedules other activities and elements. It is the controller of the Model-View-Controller pattern.                          |          |
| <b>Control Flow</b>          | A connector linking two nodes in an activity diagram. Control Flow connectors start a node's activity when the preceding node's action is finished.                                                                       |          |

## 19.4 D

| Name                            | Detail                                                                                                                                                                                                                                                                                  | See also           |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <b>Database Schema</b>          | The description of a database structure. It defines tables and fields and the relationship between them.                                                                                                                                                                                |                    |
| <b>Datastore</b>                | An element used to define permanently stored data. A token of data that is stored in the Datastore is stored permanently. A token of data that comes out of the Datastore is a copy of the original data. The tokens imported are kept for the life of the Activity in which it exists. |                    |
| <b>Datatype</b>                 | A descriptor of a set of values that lack identity and whose operations do not have side effects. Datatypes include primitive pre-defined types and user-definable types. Pre-defined types include numbers, string and time. User-definable types include enumerations.                |                    |
| <b>Decision</b>                 | An element of an Activity diagram that indicates a point of conditional progression: if a condition is true, then processing continues one way, if not, then another.                                                                                                                   |                    |
| <b>Defining Model (MOF)</b>     | The model on which a repository is based. Any number of repositories can have the same defining model.                                                                                                                                                                                  |                    |
| <b>Delegate</b>                 | A connector that defines the internal assembly of a component's external ports and interfaces. Using a Delegate connector wires the internal workings of the system to the outside world, by a delegation of the external interfaces' connections.                                      |                    |
| <b>Delegation</b>               | The ability of an object to issue a message to another object in response to a message. Delegation can be used as an alternative to inheritance.<br><br>Contrast: inheritance                                                                                                           |                    |
| <b>Dependency</b>               | A relationship between two modeling elements, in which a change to one modeling element (the independent element) affects the other modeling element (the dependent element).                                                                                                           |                    |
| <b>Deployment</b>               | A type of Dependency relationship that indicates the deployment of an artifact onto a node or executable target.                                                                                                                                                                        |                    |
| <b>Deployment Diagram</b>       | A diagram that shows the configuration of run-time processing nodes and the components, processes, and objects that live on them. Components represent run-time manifestations of code units.                                                                                           | Component Diagrams |
| <b>Deployment Specification</b> | Specifies parameters guiding deployment of an artifact, as is common with most hardware and software technologies.                                                                                                                                                                      |                    |
| <b>Derived Element</b>          | A model element that can be computed from another element, but that is shown for clarity or that is included for design purposes even though it adds no semantic information.                                                                                                           |                    |
| <b>Design</b>                   | The part of the software development process whose primary purpose is to decide how the system is to be                                                                                                                                                                                 | Analysis           |

| Name                          | Detail                                                                                                                                                                                                                                                                                                                         | See also                                             |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
|                               | implemented. During design, strategic and tactical decisions are made to meet the required functional and quality requirements of a system.                                                                                                                                                                                    |                                                      |
| <b>Design Time</b>            | Refers to something that occurs during a design phase of the software development process.                                                                                                                                                                                                                                     | Modeling Time, Run Time, Compile Time, Analysis Time |
| <b>Development Process</b>    | A set of partially ordered steps performed for a given purpose during software development, such as constructing models or implementing models.                                                                                                                                                                                |                                                      |
| <b>Diagram</b>                | A graphical presentation of a collection of model elements, most often rendered as a connected graph of arcs (relationships) and vertices (other model elements). UML supports 14 diagram types, and Enterprise Architect extends these with seven more. Add-Ins, technologies and profiles can provide further diagram types. |                                                      |
| <b>Diagram Gate</b>           | A simple graphical way to indicate the point at which messages can be transmitted into and out of Interaction Fragments.                                                                                                                                                                                                       |                                                      |
| <b>Diagram View</b>           | The Enterprise Architect workspace area where the UML diagrams are displayed.                                                                                                                                                                                                                                                  |                                                      |
| <b>Disjoint Substate</b>      | A substate that cannot be held simultaneously with other substates contained in the same composite State.                                                                                                                                                                                                                      | Composite State, Substate, Concurrent Substate       |
| <b>Distribution Unit</b>      | A set of objects or components that are allocated to a process or a processor as a group. A distribution unit can be represented by a run-time composite or an Aggregate.                                                                                                                                                      |                                                      |
| <b>Domain</b>                 | An area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area.                                                                                                                                                                                                  |                                                      |
| <b>Dynamic Classification</b> | A semantic variation of Generalization in which an object can change its classifier.                                                                                                                                                                                                                                           | Multiple Classification, Static Classification       |

## 19.5 E

| Name                     | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | See also            |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <b>Element</b>           | <ol style="list-style-type: none"> <li>1. An atomic constituent of a model.</li> <li>2. A model object of any type, such as Class, Component, Node or Object.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                           |                     |
| <b>Endpoint</b>          | Used in Interaction diagrams to reflect a lost message in sequence.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                     |
| <b>Entity</b>            | A store or persistence mechanism that captures the information or knowledge in a system. It is the Model in the Model-View-Controller pattern.                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |
| <b>Entry Action</b>      | An action executed upon entering a state in a State Machine regardless of the transition taken to reach that state.                                                                                                                                                                                                                                                                                                                                                                                                                                                |                     |
| <b>Entry Point</b>       | Used to define where external states can enter a Sub Machine.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                     |
| <b>Enumeration</b>       | A list of named values used as the range of a particular attribute type. For example, RGBColor = {red, green, blue}. Boolean is a predefined enumeration with values from the set {false, true}.                                                                                                                                                                                                                                                                                                                                                                   |                     |
| <b>Event</b>             | The specification of a significant occurrence that has a location in time and space. In the context of State diagrams, an event is an occurrence that can trigger a transition.                                                                                                                                                                                                                                                                                                                                                                                    |                     |
| <b>Exception Handler</b> | An element that defines the group of operations to carry out when an exception occurs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                     |
| <b>Exit Action</b>       | An action executed upon exiting a State in a State Machine regardless of the transition taken to exit that State.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                     |
| <b>Exit Point</b>        | Used in Sub Machine states and State Machines to denote the point where the machine is exited and the transition sourcing this exit point, for Sub Machines, is triggered. Exit points are a type of pseudo-state used in the State Machine diagram.                                                                                                                                                                                                                                                                                                               |                     |
| <b>Export</b>            | In the context of packages, to make an element visible outside its enclosing namespace.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Visibility , Import |
| <b>Expose Interface</b>  | A toolbox icon that is a graphical way to depict the required and supplied interfaces of a Component, Class or Part.                                                                                                                                                                                                                                                                                                                                                                                                                                               |                     |
| <b>Expression</b>        | A string that evaluates to a value of a particular type. For example, the expression $(7 + 5 * 3)$ evaluates to a value of type number. A relationship from an extension Use Case to a base Use Case, specifying how the behavior defined for the extension Use Case augments (subject to conditions specified in the extension) the behavior defined for the base Use Case. The behavior is inserted at the location defined by the extension point in the base Use Case. The base Use Case does not depend on performing the behavior of the extension Use Case. | Extend , Include    |

---

| Name          | Detail                                                                                                                                                                                      | See also             |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| <b>Extend</b> | A connector used to indicate that an element extends the behavior of another. Extensions are used in Use Case models to indicate one Use Case (optionally) extends the behavior of another. | Expression , Include |

## 19.6 F

| Name                       | Detail                                                                                                                                                                                                                                                                                                           | See also        |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <b>Facade</b>              | A stereotyped package containing only references to model elements owned by another package. It is used to provide a 'public view' of some of the contents of a package.                                                                                                                                         |                 |
| <b>Feature</b>             | A property, like operation or attribute, that is encapsulated within a classifier, such as an Interface, a Class, or a Datatype.                                                                                                                                                                                 |                 |
| <b>Final</b>               | A pseudo-state that indicates an end.                                                                                                                                                                                                                                                                            |                 |
| <b>Final State</b>         | A special kind of State signifying that the enclosing composite State or the entire State Machine is completed.                                                                                                                                                                                                  |                 |
| <b>Fire</b>                | To execute a State transition.                                                                                                                                                                                                                                                                                   | Transition      |
| <b>Flow Final</b>          | An element that depicts an exit from the system, as opposed to the Activity Final, which represents the completion of the activity.                                                                                                                                                                              |                 |
| <b>Focus Class</b>         | A stereotyped Class that defines the core logic or control flow for one or more auxiliary Classes that support it. Focus Classes are typically used together with one or more auxiliary Classes, and are particularly useful for specifying the core business logic or control flow of components during design. | Auxiliary Class |
| <b>Focus of Control</b>    | A symbol on a Sequence diagram that shows the period of time during which an object is performing an action, either directly or through a subordinate procedure.                                                                                                                                                 |                 |
| <b>Forward Engineering</b> | The process of generating source code from the UML model.                                                                                                                                                                                                                                                        |                 |
| <b>Fork</b>                | Used in State Machine diagrams as pseudo-states. With respect to State Machine diagrams, a Fork pseudo-state signifies that its incoming transition comes from a single State, and it has multiple outgoing transitions.                                                                                         | Join            |
| <b>Framework</b>           | A stereotyped package containing model elements that specify a reusable architecture for all or part of a system. Frameworks typically include Classes, Patterns or templates. When frameworks are specialized for an application domain, they are sometimes referred to as Application frameworks.              | Pattern         |

## 19.7 G

| Name                         | Detail                                                                                                                                                                                                                                                                                                | See also                           |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| <b>Generalizable Element</b> | A model element that can participate in a Generalization relationship.                                                                                                                                                                                                                                | Generalization                     |
| <b>Generalization</b>        | A taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element can be used where the more general element is allowed. | Generalizable Element, Inheritance |
| <b>Guard Condition</b>       | A condition that must be satisfied in order to enable an associated transition to fire.                                                                                                                                                                                                               |                                    |

**19.8 H**

| Name                 | Detail                                                                                                                                                                                                                                                                                                                | See also |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>History State</b> | There are two types of History pseudo-states defined in UML: shallow History and deep History. A shallow History sub-state is used to represent the most recently active sub-state of a composite State. A deep History sub-state, in contrast, reflects the most recent active configuration of the composite State. |          |

## 19.9 I

| Name                                | Detail                                                                                                                                                                                                                                                                                                                                                                                        | See also                   |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| <b>Implementation</b>               | A definition of how something is constructed or computed. For example, a Class is an implementation of a type, a method is an implementation of an operation.                                                                                                                                                                                                                                 |                            |
| <b>Implementation Class</b>         | A stereotyped Class that specifies the implementation of a Class in some programming language (for example, C++, Smalltalk, Java) in which an instance can not have more than one Class. An Implementation Class is said to realize a type if it provides all of the operations defined for the type with the same behavior as specified for the type's operations.                           | Type                       |
| <b>Implementation Inheritance</b>   | The inheritance of the implementation of a more general element. Includes inheritance of the interface.                                                                                                                                                                                                                                                                                       | Interface Inheritance      |
| <b>Import</b>                       | In the context of packages, a dependency that shows the packages whose Classes can be referenced within a given package (including packages recursively embedded within it).                                                                                                                                                                                                                  | Visibility , Export        |
| <b>Include</b>                      | A relationship from a base Use Case to an inclusion Use Case, specifying how the behavior for the base Use Case contains the behavior of the inclusion Use Case. The behavior is included at the location that is defined in the base Use Case. The base Use Case depends on performing the behavior of the inclusion Use Case, but not on its structure (that is, attributes or operations). | Extend, Expression         |
| <b>Inheritance</b>                  | The mechanism by which more specific elements incorporate the structure and behavior of more general elements related by behavior.                                                                                                                                                                                                                                                            | Generalization, Delegation |
| <b>Initial State</b>                | A pseudo-state used to denote the default state of a composite State; there can be one initial vertex in each region of the composite State.                                                                                                                                                                                                                                                  |                            |
| <b>Instance</b>                     | An entity that has a unique identity, a set of operations that can be applied to it, and a state that stores the effects of the operations.                                                                                                                                                                                                                                                   | Object                     |
| <b>Interaction</b>                  | A specification of how stimuli are sent between instances to perform a specific task. The interaction is defined in the context of a collaboration.                                                                                                                                                                                                                                           | Collaboration              |
| <b>Interaction Diagram</b>          | A generic term that applies to several types of diagrams that emphasize object interactions. These include Timing diagrams, Sequence diagrams, Interaction Overview diagrams and Communication diagrams.                                                                                                                                                                                      |                            |
| <b>Interaction Occurrence</b>       | A reference to an existing interaction element. Interaction occurrences are visually represented by a frame, with ref in the frame's title space. The diagram name is indicated in the frame contents.                                                                                                                                                                                        |                            |
| <b>Interaction Overview Diagram</b> | A diagram that visualizes the cooperation between other Interaction diagrams to illustrate a control flow serving an encompassing purpose. As Interaction Overview diagrams are a variant of Activity diagrams, most of the                                                                                                                                                                   |                            |

| Name                         | Detail                                                                                                                        | See also                   |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------|----------------------------|
|                              | diagram notation is similar, as is the process in constructing the diagram.                                                   |                            |
| <b>Interface</b>             | A named set of operations that characterize the behavior of an element.                                                       | Class , Type               |
| <b>Interface Inheritance</b> | The inheritance of the interface of a more general element. Does not include inheritance of the implementation.               | Implementation Inheritance |
| <b>Internal Transition</b>   | A transition signifying a response to an event without changing the state of an object.                                       |                            |
| <b>Interrupt Flow</b>        | An Enterprise Architect-defined toolbox icon used to define the exception handler and interruptible activity region concepts. |                            |

## 19.10 J

| Name            | Detail                                                                                                                                                                                                            | See also |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| <b>Java</b>     | A fully object-oriented, cross platform language based on elements from Smalltalk, C++ and other OO languages.                                                                                                    |          |
| <b>Join</b>     | Used in State Machine diagrams and in Activity diagrams to synchronize multiple flows.                                                                                                                            | Fork     |
| <b>Junction</b> | Junction pseudo-states are used to design complex transitional paths. A Junction can be used to combine, or merge, multiple paths into a shared transition path or to split an incoming path into multiple paths. |          |

**19.11 L**

| <b>Name</b>       | <b>Detail</b>                                                                                                                                                                                 | <b>See also</b>                |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <b>Layer</b>      | The organization of classifiers or packages at the same level of abstraction. A layer represents a horizontal slice through an architecture, whereas a partition represents a vertical slice. | Partition                      |
| <b>Lifeline</b>   | An individual participant in an interaction (that is, Lifelines cannot have multiplicity). A Lifeline represents a distinct connectable element.                                              |                                |
| <b>Link</b>       | A semantic connector among a tuple of objects. An instance of an Association.                                                                                                                 | Association                    |
| <b>Link End</b>   | An instance of an Association end.                                                                                                                                                            | Association End,<br>Classifier |
| <b>Local Path</b> | A relative path on a local machine, enabling developers to store shared source code in machine specific directories, but still generate and synchronize code.                                 |                                |

## 19.12 M

| Name                              | Detail                                                                                                                                                                                                                                       | See also        |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <b>Maintenance</b>                | The support of a software system after it is deployed.                                                                                                                                                                                       |                 |
| <b>Manifest</b>                   | A relationship that indicates that the artifact source embodies the target model element. Stereotypes can be added to Enterprise Architect to classify the type of manifestation of the model element.                                       |                 |
| <b>Message</b>                    | Messages indicate a flow of information, or transition of control, between elements. Messages are used by Communication diagrams, Sequence diagrams, Interaction Overview diagrams and Timing diagrams.                                      |                 |
| <b>Message Endpoint</b>           | An element that defines an endpoint of a Lifeline, such as a State or Value Lifeline in a Timing diagram.                                                                                                                                    |                 |
| <b>Message Label</b>              | Used for messages sent between Lifelines to make the diagram appear less cluttered. Labels with the same name indicate that a message can be interrupted.                                                                                    |                 |
| <b>Metaclass</b>                  | A Class whose instances are Classes. Metaclasses are typically used to construct metamodels                                                                                                                                                  |                 |
| <b>Metafile</b>                   | A vector-based image format native to Windows. Supports high detail and excellent scaling. Typically used for saving diagram images for placement in documents. Comes in Placeable (an older format) and Enhanced (current standard format). |                 |
| <b>Meta-Metamodel</b>             | A model that defines the language for expressing a metamodel. The relationship between a meta-metamodel and a metamodel is analogous to the relationship between a metamodel and a model.                                                    |                 |
| <b>Metamodel</b>                  | A model that defines the language for expressing a model.                                                                                                                                                                                    |                 |
| <b>Meta-Object</b>                | A generic term for all meta-entities in a meta-modeling language. For example, meta-types, meta-classes, meta-attributes, and meta-associations.                                                                                             |                 |
| <b>Meta-Object Facility (MOF)</b> | An Object Management Group (OMG) standard. MOF originated in the UML, when the OMG required a Meta-Modeling architecture to define the UML. MOF is designed as a four-layered architecture.                                                  |                 |
| <b>Method</b>                     | The implementation of an operation. It specifies the algorithm or procedure associated with an operation.                                                                                                                                    |                 |
| <b>Model ( MOF )</b>              | An abstraction of a physical system with a certain purpose. Usage note: In the context of the MOF specification, which describes a meta-metamodel, the meta-metamodel is for brevity frequently referred to simply as the model.             | Physical System |
| <b>Model Aspect</b>               | A dimension of modeling that emphasizes particular qualities of the metamodel. For example, the structural model aspect emphasizes the structural qualities of the metamodel.                                                                |                 |

| Name                           | Detail                                                                                                                                                                                                                                                                                                                  | See also                                           |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>Model Elaboration</b>       | The process of generating a repository type from a published model. Includes the generation of interfaces and implementations which enables repositories to be instantiated and populated based on, and in compliance with, the model elaborated.                                                                       |                                                    |
| <b>Model Element ( MOF )</b>   | An element that is an abstraction drawn from the system being modeled. In the MOF specification model elements are considered to be meta-objects.                                                                                                                                                                       | View Element;                                      |
| <b>Model Library</b>           | A stereotyped package containing model elements that are intended to be reused by other packages. A model library differs from a profile in that a model library does not extend the metamodel using stereotypes and tagged definitions. A model library is analogous to a Class library in some programming languages. |                                                    |
| <b>Modeling Time</b>           | Refers to something that occurs during the modeling phase of the software development process. It includes analysis time and design time. Usage note: When discussing object systems, it is often important to distinguish between modeling-time and run-time concerns.                                                 | Analysis Time, Design Time, Compile time, Run time |
| <b>Module</b>                  | A software unit of storage and manipulation. Modules include source code modules, binary code modules and executable code modules.                                                                                                                                                                                      | Component                                          |
| <b>MOF</b>                     | Meta-Object Facility, an Object Management Group (OMG) standard. MOF originated in the UML, when the OMG required a Meta-Modeling architecture to define the UML. MOF is designed as a four-layered architecture.                                                                                                       |                                                    |
| <b>Multiple Classification</b> | A semantic variation of Generalization in which an object can belong directly to more than one classifier.                                                                                                                                                                                                              | Static Classification, Dynamic Classification      |
| <b>Multiple Inheritance</b>    | A semantic variation of Generalization in which a type can have more than one supertype.                                                                                                                                                                                                                                | Single Inheritance                                 |
| <b>Multiplicity</b>            | A specification of the range of enableable cardinalities that a set can assume. Multiplicity specifications can be given for roles within Associations, Parts within Composites, repetitions and other purposes. Essentially a multiplicity is a (possibly infinite) subset of the non-negative integers.               | Cardinality                                        |
| <b>Multi-Valued ( MOF )</b>    | A model element with multiplicity defined whose Multiplicity Type:: upper attribute is set to a number greater than one. The term multi-valued does not pertain to the number of values held by, for example, an attribute or parameter, at any point in time.                                                          | Single-Valued                                      |

**19.13 N**

| Name                     | Detail                                                                                                                                                                                                                                                                   | See also           |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| <b>Name</b>              | A string used to identify a model element.                                                                                                                                                                                                                               |                    |
| <b>Namespace</b>         | A part of the model in which the names can be defined and used. Within a namespace, each name has a unique meaning.                                                                                                                                                      | Name               |
| <b>N-ary Association</b> | An Association among three or more Classes. Each instance of the Association is an n-tuple of values from the respective Classes.                                                                                                                                        | Binary Association |
| <b>Nesting</b>           | A connector used as an alternative membership notation to indicate nested members within an element; for example, a package that has nested members. The nested members of a package could also be shown inside the package rather than linked by the Nesting connector. |                    |
| <b>Node</b>              | A classifier that represents a run-time computation resource, which generally has at least a memory and often processing capability. Run-time objects and components can reside on nodes.                                                                                |                    |

## 19.14 O

| Name                                 | Detail                                                                                                                                                                                                                                                  | See also                                                                                           |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>Object</b>                        | An entity with a well-defined boundary and identity that encapsulates state and behavior. State is represented by attributes and relationships, behavior is represented by operations, methods and State Machines. An Object is an instance of a Class. | Class, Instance                                                                                    |
| <b>Object Diagram</b>                | A diagram that encompasses objects and their relationships at a point in time. An Object diagram can be considered as a special case of a Class diagram or Communication diagram.                                                                       | Class Diagram, Communication Diagram                                                               |
| <b>Object Flow</b>                   | A sub type of the State flow or transition. It implies the passing of an object instance between elements at run-time.                                                                                                                                  |                                                                                                    |
| <b>Object Flow State</b>             | A state in an Activity graph that represents the passing of an object from the output of actions in one State to the input of actions in another State.                                                                                                 |                                                                                                    |
| <b>Object Lifeline</b>               | A line in a Sequence diagram that represents the existence of an object over a period of time.                                                                                                                                                          | Sequence Diagram                                                                                   |
| <b>Object Management Group (OMG)</b> | The standards body responsible for the UML specification and management.                                                                                                                                                                                | Their website is <a href="http://www.omg.org">www.omg.org</a> - follow the links to the UML pages. |
| <b>Occurrence</b>                    | A relationship that indicates that a Collaboration represents a classifier. An Occurrence connector is drawn from the collaboration to the classifier.                                                                                                  |                                                                                                    |
| <b>Operation</b>                     | A service that can be requested from an object to effect behavior. An operation has a signature, which could restrict the actual parameters that are possible.                                                                                          |                                                                                                    |

## 19.15 P

| Name                   | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | See also                                          |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <b>Package</b>         | <p>1. A namespace, as well as an element that can be contained in other packages' namespaces. Packages can own or merge with other packages, and their elements can be imported into a package's namespace.</p> <p>2. A logical container of model elements. It groups elements and can also contain other packages.</p> <p>The OMG UML specification (UML Superstructure Specification, v2.1.1, p. 109) states:</p> <p>A package is used to group elements, and provides a namespace for the grouped elements.</p> <p>A package is a namespace for its members, and can contain other packages. Only packageable elements can be owned members of a package. By virtue of being a namespace, a package can import either individual members of other packages, or all the members of other packages.</p> <p>In addition a package can be merged with other packages.</p> <p>Note that packages own model elements and are the basis for configuration control, storage and access control. Each element can be directly owned by a single package, so the package hierarchy is a strict tree. However, packages can reference other packages, modeled by using one of the stereotypes «import» and «access» of Permission dependency, so the usage network is a graph. Other kinds of dependencies between packages usually imply that one or more dependencies among the elements exist.</p> <p>A package is represented by the common folder icon - a large rectangle with a small rectangle (a 'tab') attached to the left side on top.</p> |                                                   |
| <b>Package Diagram</b> | Used to reflect the organization of packages and their elements, and provide a visualization of their corresponding namespaces.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                   |
| <b>Package Import</b>  | A package import relationship is drawn from a source package to a package whose contents are imported. Private members of a target package cannot be imported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                   |
| <b>Package Merge</b>   | Indicates a relationship between two packages whereby the contents of the target package are merged with those of the source package. Private contents of a target package are not merged.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                   |
| <b>Parameter</b>       | The specification of a variable that can be changed, passed, or returned. A parameter can include a name, type, and direction. Parameters are used for operations, messages and events.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Formal Parameter ,<br>Argument , Actual parameter |
| <b>Parameterized</b>   | The descriptor for a Class with one or more unbound                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Template, Parameterized                           |

| Name                     | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                   | See also                 |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| <b>element</b>           | parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                              | class                    |
| <b>Parent</b>            | In a generalization relationship, the generalization of another element, the child.                                                                                                                                                                                                                                                                                                                                                                      | Subclass, Subtype, Child |
| <b>Part</b>              | A run-time instance of a Class or Interface.                                                                                                                                                                                                                                                                                                                                                                                                             |                          |
| <b>Participate</b>       | The connection of a model element to a relationship or to a reified relationship. For example, a Class participates in an Association, an Actor participates in a Use Case.                                                                                                                                                                                                                                                                              |                          |
| <b>Partition</b>         | <ol style="list-style-type: none"> <li>1. activity graphs: A portion of an activity graph that organizes the responsibilities for actions.</li> <li>2. architecture: A set of related classifiers or packages at the same level of abstraction or across layers in a layered architecture. A partition represents a vertical slice through an architecture, whereas a layer represents a horizontal slice.</li> </ol>                                    | Layer , Swim Lane        |
| <b>Pattern</b>           | A template collaboration.                                                                                                                                                                                                                                                                                                                                                                                                                                | Framework                |
| <b>Persistent Object</b> | An object that exists after the process or thread that created it has ceased to exist.                                                                                                                                                                                                                                                                                                                                                                   |                          |
| <b>Physical System</b>   | <ol style="list-style-type: none"> <li>1. The subject of a model.</li> <li>2. A collection of connected physical units, which can include software, hardware and people, that are organized to accomplish a specific purpose. A physical system can be described by one or more models, possibly from different viewpoints.</li> </ol>                                                                                                                   | Model (MOF) , System     |
| <b>Port</b>              | Defines the interaction between a classifier and its environment. Interfaces controlling this interaction can be depicted using the 'Expose Interface' toolbox icon.                                                                                                                                                                                                                                                                                     |                          |
| <b>Postcondition</b>     | A constraint that must be true at the completion of an operation.                                                                                                                                                                                                                                                                                                                                                                                        |                          |
| <b>Precondition</b>      | A constraint that must be true when an operation is invoked.                                                                                                                                                                                                                                                                                                                                                                                             |                          |
| <b>Primitive Type</b>    | A pre-defined basic datatype without any substructure, such as an integer or a string.                                                                                                                                                                                                                                                                                                                                                                   |                          |
| <b>Process</b>           | <ol style="list-style-type: none"> <li>1. A heavyweight unit of concurrency and execution in an operating system. Thread, which includes heavyweight and lightweight processes. If necessary, an implementation distinction can be made using stereotypes.</li> <li>2. A software development process - the steps and guidelines by which to develop a system.</li> <li>3. To execute an algorithm or otherwise handle something dynamically.</li> </ol> | Thread                   |
| <b>Product</b>           | A physical piece of information that is produced by a business or development process. Examples of products include models, source files, scripts, and binary executable files. An product can constitute the                                                                                                                                                                                                                                            | Artifact , Component     |

| Name                         | Detail                                                                                                                                                                                                                                                                                                      | See also     |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
|                              | implementation of a deployable component.                                                                                                                                                                                                                                                                   |              |
| <b>Profile</b>               | A stereotyped package that contains model elements that have been customized for a specific domain or purpose using extension mechanisms, such as stereotypes, tagged definitions and constraints. A profile can also specify model libraries on which it depends and the metamodel subset that it extends. |              |
| <b>Project Browser</b>       | The workspace window where the model contents are displayed in 'tree' format. Displays structures such as packages, diagrams and model elements.                                                                                                                                                            |              |
| <b>Projection</b>            | A mapping from a set to its subset.                                                                                                                                                                                                                                                                         |              |
| <b>Property</b>              | A named value denoting a characteristic of an element. A property has semantic impact. Certain properties are predefined in the UML; others can be user defined.                                                                                                                                            | Tagged Value |
| <b>Pseudo-State</b>          | A vertex in a State Machine that has the form of a State, but doesn't behave as a State. Pseudo-states include initial and history vertices.                                                                                                                                                                |              |
| <b>Published Model (MOF)</b> | A model that has been frozen, and that becomes available for instantiating repositories and for support in defining other models. A frozen model's model elements cannot be changed.                                                                                                                        |              |

**19.16 Q**

| Name      | Detail                                                                                                                                | See also |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------|----------|
| Qualifier | An Association attribute or tuple of attributes whose values partition the set of objects related to an object across an Association. |          |

## 19.17 R

| Name                         | Detail                                                                                                                                                                                                                                                                                                                                                                                                                 | See also         |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <b>Realize</b>               | A source object realizes the destination object. Realize is used to express traceability and completeness in the model – a business process or requirement is realized by one or more Use Cases which are in turn realized by some Classes which in turn are realized by a Component, and so on.                                                                                                                       |                  |
| <b>Receive ( A message )</b> | The handling of a stimulus passed from a sender instance.                                                                                                                                                                                                                                                                                                                                                              | Sender, Receiver |
| <b>Receive</b>               | An element used to define the acceptance or receipt of a request. Movement on to the next action occurs until it has received what is defined.                                                                                                                                                                                                                                                                         |                  |
| <b>Receiver ( Object )</b>   | The object handling a stimulus passed from a sender object.                                                                                                                                                                                                                                                                                                                                                            | Sender           |
| <b>Reception</b>             | A declaration that a classifier is prepared to react to the receipt of a signal.                                                                                                                                                                                                                                                                                                                                       |                  |
| <b>Recursion</b>             | A type of message used in Sequence diagrams to indicate a recursive function.                                                                                                                                                                                                                                                                                                                                          |                  |
| <b>Reference</b>             | 1. A denotation of a model element.<br>2. A named slot within a classifier that facilitates navigation to other classifiers.                                                                                                                                                                                                                                                                                           | Pointer          |
| <b>Region</b>                | UML 2.x supports both Expansion Regions and Interruptible Activity Regions. An Expansion Region defines the bounds of a region consisting of one or more sets of input collections, where an input collection is a set of elements of the same type. An Interruptible Activity Region contains Activity nodes - when a token leaves an interruptible region, this terminates all of the region's tokens and behaviors. |                  |
| <b>Refinement</b>            | A relationship that represents a fuller specification of something that has already been specified at a certain level of detail. For example, a design Class is a refinement of an analysis Class.                                                                                                                                                                                                                     |                  |
| <b>Relationship</b>          | A semantic connection among model elements. Examples of relationships include Associations and Generalizations.                                                                                                                                                                                                                                                                                                        |                  |
| <b>Repository</b>            | A facility for storing object models, interfaces and implementations.                                                                                                                                                                                                                                                                                                                                                  |                  |
| <b>Represents</b>            | A connector that indicates a Collaboration Use is used in a classifier. The connector is drawn from the Collaboration Use to its owning classifier.                                                                                                                                                                                                                                                                    |                  |
| <b>Requirement</b>           | A required feature, property or behavior of a system (external requirement).                                                                                                                                                                                                                                                                                                                                           |                  |
| <b>Responsibility</b>        | A contract or obligation of a classifier (internal requirement).                                                                                                                                                                                                                                                                                                                                                       |                  |

| Name                       | Detail                                                                                                                                                                                                                                                                                                                                                                                   | See also                                                                                                                                 |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Reuse</b>               | The use of a pre-existing artifact.                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                          |
| <b>Reverse Engineering</b> | The process of importing source code into the model as standard UML model objects (such as Classes, attributes and operations).                                                                                                                                                                                                                                                          |                                                                                                                                          |
| <b>Rich Text Format</b>    | A standard mark-up language for creating word processor documents, frequently associated with Microsoft Word.                                                                                                                                                                                                                                                                            |                                                                                                                                          |
| <b>Robustness Diagram</b>  | Enterprise Architect supports business process modeling extensions from the UML business process model profile. Robustness diagrams are used in ICONIX.                                                                                                                                                                                                                                  | You can read more about this at <a href="http://www.sparxsystems.com/iconix/iconixsw.htm">www.sparxsystems.com/iconix/iconixsw.htm</a> . |
| <b>Role</b>                | <ol style="list-style-type: none"> <li>1. The named detail and rules associated with one end of an association. Can indicate name, constraints, multiplicity and collection details.</li> <li>2. The named specific behavior of an entity participating in a particular context. A role can be static (such as an Association end) or dynamic (such as a Collaboration role).</li> </ol> |                                                                                                                                          |
| <b>Role Binding</b>        | The mapping between a Collaboration Use's internal roles and the respective parts required to implement a specific situation. The associated parts can have properties defined to enable the binding to occur, and the collaboration to take place.                                                                                                                                      |                                                                                                                                          |
| <b>Run Time</b>            | The period of time during which a computer program executes.                                                                                                                                                                                                                                                                                                                             | Analysis Time, Compile Time, Design Time, Modeling Time                                                                                  |

## 19.18 S

| Name                            | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | See also                               |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| <b>Scenario</b>                 | <ol style="list-style-type: none"> <li>1. A specific sequence of actions that illustrates behaviors. A scenario can be used to illustrate an interaction or the execution of a Use Case instance.</li> <li>2. A sequence of operations carried out in some order to produce a known result. Can apply to Use Cases where it is the equivalent of a Sequence diagram, or to other objects to describe how they are used at run-time.</li> </ol>                                                                                                                             | Interaction                            |
| <b>Schema ( MOF )</b>           | In the context of the MOF, analogous to a package that is a container of model elements. Schema corresponds to a MOF package.                                                                                                                                                                                                                                                                                                                                                                                                                                              | Metamodel, Package                     |
| <b>Self-Message</b>             | Reflects a new process or method invoked within the calling Lifeline's operation. It is a specification of a message.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                        |
| <b>Semantic Variation Point</b> | A point of variation in the semantics of a metamodel. It provides an intentional degree of freedom for the interpretation of the metamodel semantics.                                                                                                                                                                                                                                                                                                                                                                                                                      |                                        |
| <b>Send ( A message )</b>       | The passing of a stimulus from a sender instance to a receiver instance.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Sender, Receiver                       |
| <b>Sender ( Object )</b>        | The object passing a stimulus to a receiver object.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Receiver                               |
| <b>Sequence Diagram</b>         | A diagram that shows object interactions arranged in time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged. Unlike a Communication (Collaboration) diagram, a Sequence diagram includes time sequences but does not include object relationships. A Sequence diagram can exist in a generic form (describes all possible scenarios) and in an instance form (describes one actual scenario). Sequence diagrams and Communication diagrams express similar information, but show it in different ways. | Communication Diagram, Object Lifeline |
| <b>Signal</b>                   | The specification of an asynchronous stimulus communicated between instances. Signals can have parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                        |
| <b>Signature</b>                | The name and parameters of a behavioral feature. A signature can include an optional returned parameter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                        |
| <b>Single Inheritance</b>       | A semantic variation of Generalization in which a type can have only one supertype.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Multiple Inheritance                   |
| <b>Single Valued ( MOF )</b>    | A model element with multiplicity defined is single valued when its Multiplicity Type: upper attribute is set to 1. The term single-valued does not pertain to the number of values held by, for example, an attribute or parameter at any point in time, since a single-valued attribute (for instance, with a multiplicity lower bound of zero) could have no value.                                                                                                                                                                                                     | Multi-Valued                           |
| <b>Specification</b>            | A declarative description of what something is or does.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Implementation                         |

| Name                         | Detail                                                                                                                                                                                                                                                                                                                                                                                               | See also                                         |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| <b>State</b>                 | A condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event.                                                                                                                                                                                                                                                     | State ( OMA )                                    |
| <b>State Invariant</b>       | A condition applied to a Lifeline that must be fulfilled for the Lifeline to exist.                                                                                                                                                                                                                                                                                                                  |                                                  |
| <b>State Machine</b>         | A behavior that specifies the sequences of States that an object or an interaction goes through during its life in response to events, together with its responses and actions.                                                                                                                                                                                                                      |                                                  |
| <b>State Machine Diagram</b> | A diagram that illustrates how an element, often a Class, can move between States classifying its behavior, according to transition triggers, constraining guards and other aspects of State Machine diagrams that depict and explain movement and behavior.                                                                                                                                         |                                                  |
| <b>State Chart</b>           | A diagram that shows a State Machine.                                                                                                                                                                                                                                                                                                                                                                | State Machine , Activity Graph                   |
| <b>State Continuation</b>    | A symbol that serves two different purposes for Interaction diagrams - as State Invariants and as Continuations. A State Invariant is a condition applied to a Lifeline that must be fulfilled for the Lifeline to exist. A Continuation is used in seq and alt combined fragments to indicate the branches of continuation that an operand follows.                                                 |                                                  |
| <b>State Lifeline</b>        | A State Lifeline follows discrete transitions between States, which are defined along the y-axis of the timeline. Any transition has optional attributes of timing constraints, duration constraints and observations.                                                                                                                                                                               |                                                  |
| <b>Static Classification</b> | A semantic variation of Generalization in which an object can not change classifier.                                                                                                                                                                                                                                                                                                                 | Multiple Classification , Dynamic Classification |
| <b>Stereotype</b>            | A new type of modeling element that extends the semantics of the metamodel. Stereotypes must be based on certain existing types or Classes in the metamodel. Stereotypes can extend the semantics, but not the structure of pre-existing types and Classes. Certain stereotypes are predefined in the UML, others can be user defined. Stereotypes are one of three extensibility mechanisms in UML. | Constraint , Tagged value                        |
| <b>Stimulus</b>              | The passing of information from one instance to another, such as raising a signal or invoking an operation. The receipt of a signal is normally considered an event.                                                                                                                                                                                                                                 | Message                                          |
| <b>String</b>                | A sequence of text characters. The details of string representation depend on implementation, and can include character sets that support international characters and graphics.                                                                                                                                                                                                                     |                                                  |
| <b>Structural Diagram</b>    | A diagram that depicts the structural elements composing a system or function. These diagrams can reflect the static relationships of a structure, as do Class or Package diagrams, or run-time architectures, such as Object or Composite Structure diagrams. Structural diagrams include Class diagrams, Composite Structure diagrams,                                                             |                                                  |

| Name                           | Detail                                                                                                                                                                                                                                                | See also                                                  |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
|                                | Component diagrams, Deployment diagrams, Object diagrams and Package diagrams.                                                                                                                                                                        |                                                           |
| <b>Structural Feature</b>      | A static feature of a model element, such as an attribute.                                                                                                                                                                                            |                                                           |
| <b>Structural Model Aspect</b> | A model aspect that emphasizes the structure of the objects in a system, including their types, Classes, relationships, attributes and operations.                                                                                                    |                                                           |
| <b>Subactivity State</b>       | A State in an activity graph that represents the execution of a non-atomic sequence of steps that has some duration.                                                                                                                                  |                                                           |
| <b>Subclass</b>                | In a Generalization relationship, the specialization of another Class; the superclass.                                                                                                                                                                | Generalization, Child , Parent Superclass                 |
| <b>Submachine State</b>        | A State in a State Machine that is equivalent to a composite State but its contents are described by another State Machine.                                                                                                                           |                                                           |
| <b>Subpackage</b>              | A package that is contained in another package.                                                                                                                                                                                                       |                                                           |
| <b>Substate</b>                | A State that is part of a composite State.                                                                                                                                                                                                            | Composite State , Concurrent Substate , Disjoint Substate |
| <b>Subsystem</b>               | A grouping of model elements that represents a behavioral unit in a physical system. A subsystem offers interfaces and has operations. In addition, the model elements of a subsystem can be partitioned into specification and realization elements. | Package, Physical System                                  |
| <b>Subtype</b>                 | In a Generalization relationship, the specialization of another type; the supertype.                                                                                                                                                                  | Generalization , Child , Parent Supertype                 |
| <b>Superclass</b>              | In a Generalization relationship, the generalization of another Class; the subclass.                                                                                                                                                                  | Generalization Subclass                                   |
| <b>Supertype</b>               | In a Generalization relationship, the generalization of another type; the subtype.                                                                                                                                                                    | Generalization Subtype                                    |
| <b>Supplier</b>                | A classifier that provides services that can be invoked by others.                                                                                                                                                                                    | Client                                                    |
| <b>Swimlane</b>                | A partition on an Activity diagram for organizing the responsibilities for actions. Swimlanes typically correspond to organizational units in a business model.                                                                                       | Partition                                                 |
| <b>Synch</b>                   | A State used for indicating that concurrent paths of a State Machine are synchronized. After bringing the paths to a synch state, the emerging transition indicates unison.                                                                           |                                                           |
| <b>Synchronize Code</b>        | The process of importing and exporting code changes to ensure the model and source code match.                                                                                                                                                        |                                                           |
| <b>System</b>                  | A top-level subsystem in a model.                                                                                                                                                                                                                     | Physical System                                           |
| <b>System Boundary</b>         | An element used to delineate a particular part of the system.                                                                                                                                                                                         |                                                           |

## 19.19 T

| Name                         | Detail                                                                                                                                                                                                                                                                            | See also                                   |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| <b>Table</b>                 | A relational table (composed of columns).                                                                                                                                                                                                                                         |                                            |
| <b>Tagged Value</b>          | The explicit definition of a property as a name-value pair. In a Tagged Value, the name is referred to as the tag. Certain tags are predefined in the UML; others can be user defined. Tagged Values are one of three extensibility mechanisms in UML.                            | Constraint, Property, Stereotype           |
| <b>Template</b>              | The descriptor for a Class with one or more unbound parameters.                                                                                                                                                                                                                   | Parameterized Element, Parameterized Class |
| <b>Terminate</b>             | A pseudostate indicating that upon entry of its pseudostate, the State Machine's execution ends.                                                                                                                                                                                  |                                            |
| <b>Thread ( of Control )</b> | A single path of execution through a program, a dynamic model, or some other representation of control flow. Also, a stereotype for the implementation of an active object as a lightweight process.                                                                              | Active Object, Process, Concurrency        |
| <b>Time Event</b>            | An event that denotes the time elapsed since the current state was entered.                                                                                                                                                                                                       | Event                                      |
| <b>Time Expression</b>       | An expression that resolves to an absolute or relative value of time.                                                                                                                                                                                                             |                                            |
| <b>Timing Diagram</b>        | A diagram that defines the behavior of different objects within a time-scale, with visual depictions of those objects changing state and interacting over time.                                                                                                                   |                                            |
| <b>Toolbox</b>               | The main toolbar running down the center of Enterprise Architect, from which you can select model elements to insert into diagrams. This is also known as the Toolbox and the Object toolbar.                                                                                     |                                            |
| <b>Top Level</b>             | A stereotype of package denoting the top-most package in a containment hierarchy. The topLevel stereotype defines the outer limit for looking up names, as namespaces 'see' outwards. For example, opTopLevelsubsystem represents the top of the subsystem containment hierarchy. |                                            |
| <b>Trace</b>                 | A dependency that indicates a historical or process relationship between two elements that represent the same concept without specific rules for deriving one from the other.                                                                                                     |                                            |
| <b>Transient Object</b>      | An object that exists only during the execution of the process or thread that created it.                                                                                                                                                                                         |                                            |
| <b>Transition</b>            | A relationship between two States indicating that an object in the first State performs certain specified actions and enters the second State when a specified event occurs and specified conditions are satisfied. On such a change of State, the transition is said to fire.    | Fire, Object Flow                          |
| <b>Type</b>                  | A stereotyped Class that specifies a domain of objects together with the operations applicable to the objects, without defining the physical implementation of those                                                                                                              | Implementation Class, Interface            |

---

| Name                   | Detail                                                                                                                                                                                                                                                     | See also |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|                        | objects. A type can not contain any methods, maintain its own thread of control, or be nested. However, it can have attributes and associations. Although an object can have at most one implementation Class, it can conform to multiple different types. |          |
| <b>Type Expression</b> | An expression that evaluates to a reference to one or more types.                                                                                                                                                                                          |          |

## 19.20 U

| Name                       | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | See also          |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>UML</b>                 | The Unified Modeling Language, a notation and specification for modeling software systems in an Object-Oriented manner. You can read more about UML at the OMG home page or at our UML Tutorial.                                                                                                                                                                                                                                                                                  |                   |
| <b>UML Diagrams</b>        | Diagrams used to model different aspects of the system under development. They include various elements and connectors, all of which have their own meanings and purposes. UML 2.3 includes 14 diagrams: Use Case diagram, Activity diagram, State Machine diagram, Timing diagram, Sequence diagram, Interaction Overview diagram, Communication diagram, Package diagram, Class diagram, Object diagram, Composite Structure diagram, Component diagram and Deployment diagram. |                   |
| <b>UML Toolbox</b>         | The main toolbar running down the center of Enterprise Architect from which you can select model elements to insert into diagrams. This is also known as the Toolbox and the Object toolbar.                                                                                                                                                                                                                                                                                      |                   |
| <b>Uninterpreted</b>       | A placeholder for a type or types whose implementation is not specified by the UML. Every uninterpreted value has a corresponding string representation.                                                                                                                                                                                                                                                                                                                          | Any ( CORBA )     |
| <b>Usage</b>               | A dependency in which one element (the client) requires the presence of another element (the supplier) for its correct functioning or implementation.                                                                                                                                                                                                                                                                                                                             |                   |
| <b>Use</b>                 | A connector that indicates that one element requires another to perform some interaction. The Usage relationship does not specify how the target supplier is used, other than that the source client uses it in definition or implementation.                                                                                                                                                                                                                                     |                   |
| <b>Use Case ( Class )</b>  | A UML model element that describes how a user of the proposed system interacts with the system to perform a discrete unit of work. It describes and signifies a single interaction over time that has meaning for the end user (person, machine or other system), and is required to leave the system in a complete state: either the interaction completed or was rolled back to the initial state.                                                                              | Use Case Instance |
| <b>Use Case Diagram</b>    | A diagram that captures Use Cases and Actor interactions. It describes the functional requirements of the system, the manner in which outside things (Actors) interact at the system boundary, and the response of the system.                                                                                                                                                                                                                                                    |                   |
| <b>Use Case Estimation</b> | The technique of estimating project size and complexity based on the number of Use Cases and their difficulty.                                                                                                                                                                                                                                                                                                                                                                    |                   |
| <b>Use Case Instance</b>   | The performance of a sequence of actions being specified in a Use Case. An instance of a Use Case.                                                                                                                                                                                                                                                                                                                                                                                | Use Case Class    |
| <b>Use Case Model</b>      | A model that describes a system's functional requirements in terms of Use Cases.                                                                                                                                                                                                                                                                                                                                                                                                  |                   |
| <b>Utility</b>             | A stereotype that groups global variables and procedures                                                                                                                                                                                                                                                                                                                                                                                                                          |                   |

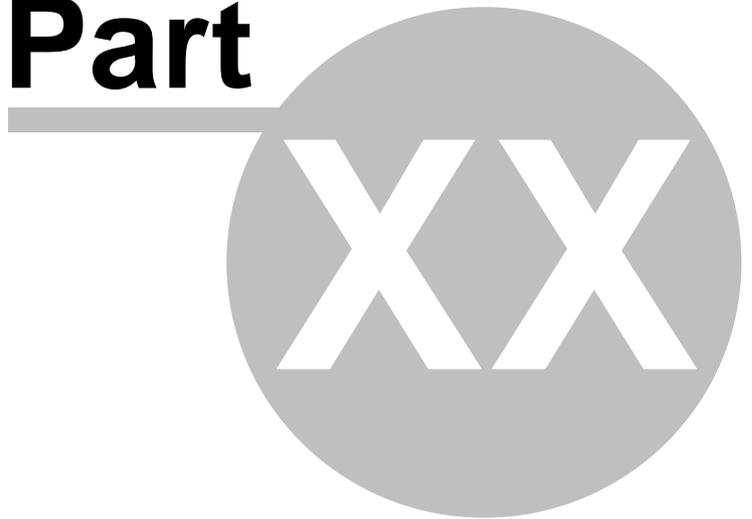
---

| Name | Detail                                                                                                                                                                                                                   | See also |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
|      | in the form of a Class declaration. The utility attributes and operations become global variables and global procedures, respectively. A utility is not a fundamental modeling construct, but a programming convenience. |          |

## 19.21 V

| Name                   | Detail                                                                                                                                                                               | See also            |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <b>Value</b>           | An element of a type domain.                                                                                                                                                         |                     |
| <b>Value Lifeline</b>  | A Lifeline that shows the Lifeline's state across the diagram, within parallel lines indicating a steady state. A cross between the lines indicates a transition or change in state. |                     |
| <b>Vertex</b>          | A source or a target for a transition in a State Machine. A vertex can be either a State or a pseudo-state.                                                                          | State, Pseudo-State |
| <b>View</b>            | A projection of a model, which is seen from a given perspective or vantage point and omits entities that are not relevant to this perspective.                                       |                     |
| <b>View Element</b>    | An element that is a textual and/or graphical projection of a collection of model elements.                                                                                          | Model Element (MOF) |
| <b>View Projection</b> | A projection of model elements onto view elements. A view projection provides a location and a style for each view element.                                                          |                     |
| <b>Visibility</b>      | An enumeration whose value (public, protected, package or private) denotes how the model element to which it refers can be seen outside its enclosing namespace                      | Export, Import      |
| <b>Visual Basic</b>    | A rapid application development programming language. Windows' only scripting language based on COM.                                                                                 |                     |

**Part**



## 20 License Management



The License Management dialog in Enterprise Architect enables you to upgrade Enterprise Architect and to register Add-Ins. The dialog lists:

- The currently-registered keys
- The product each key applies to
- When each private key expires
- When each shared key is to be reactivated in the keystore for issue to another user

**Access:** [Help | Register and Manage License Key\(s\)](#)

### Reference:

The options on the License Management dialog are explained in the following table:

| Option                                  | Action                                                                                                                                                                                                                                                                                                                                                                           | See also                                                                                                                                                                                                                           |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Add Key</b>                          | Display the Add Registration Key dialog, which enables you to: <ul style="list-style-type: none"> <li>• Register your new Enterprise Architect license</li> <li>• Obtain a shared key from the Sparx Systems Key Store (available for version 4.51 and above)</li> <li>• Add a new key to update to a higher version of Enterprise Architect or to register an Add-In</li> </ul> | <a href="#">Register a Full License</a> <sup>[209]</sup><br><a href="#">Add License Key</a> <sup>[209b]</sup><br><a href="#">Upgrade an Existing License</a> <sup>[209a]</sup><br><a href="#">Register Add-In</a> <sup>[210]</sup> |
| <b>Remove Key</b><br><b>Release Key</b> | <ul style="list-style-type: none"> <li>• (Private Key) Make the Add-In or current version of Enterprise Architect inoperable</li> <li>• (Shared Key) Release the key to the keystore; however, the Keystore Manager is normally configured to release keys automatically when the user logs off</li> </ul>                                                                       |                                                                                                                                                                                                                                    |
| <b>Copy</b>                             | Place the highlighted key into the clipboard                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                    |
| <b>Close</b>                            | Close the dialog                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                    |
| <b>Help</b>                             | Display the help for this topic                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                    |

### Learn More:

- [Upgrade an Existing License](#)<sup>[209a]</sup>
- [Register an Add-In](#)<sup>[210]</sup>

## 20.1 Finding Your License Information

You can find information on your Enterprise Architect license in the About Enterprise Architect dialog.

**Access:** [Help](#) | [About EA](#)

## 20.2 Adding License Keys

Two types of license key can be used in conjunction with Enterprise Architect: Private and Shared.

### Reference:

| Topic                | Detail                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                    |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| <b>Private Keys</b>  | <p>Private keys enable you to register, indefinitely, on the machine and user account that you are currently using:</p> <ul style="list-style-type: none"> <li>• An Enterprise Architect license for the edition of Enterprise Architect you have purchased, or</li> <li>• An Add-In product license (such as for MDG Link for Eclipse or MDG Link for Visual Studio.NET)</li> </ul>                                                                                                                                                                                                                                                                                                                                                      | <a href="#">Enterprise Architect Editions</a> <sup>13</sup> |
| <b>Shared Keys</b>   | <p>Shared keys are obtained from a central shared key store, to give you temporary access to the installed 'suite' edition of Enterprise Architect</p> <p>Shared Keys are available with the purchase of a floating license, and require Enterprise Architect version 4.51 or higher</p> <p>Shared Keys require a shared license key store to be configured by your license administrator; the key store can be either file-based or network based (preferred)</p> <p>Only the Key Administrator is required to install the Sparx Enterprise Key Store application; end users simply connect to the configured key file - advised by the administrator - as described below</p> <p>No additional software is required to be installed</p> | <a href="#">Enterprise Architect Floating License.</a>      |
| <b>Notes on Keys</b> | <p>Some license keys can override and remove others; for example:</p> <ul style="list-style-type: none"> <li>• The key for a more advanced edition of Enterprise Architect replaces the key for a simpler edition</li> <li>• The key for MDG Integration for Visual Studio replaces the key for MDG Link for Visual Studio</li> </ul> <p>Shared keys and Private keys have different formats and cannot be used interchangeably</p>                                                                                                                                                                                                                                                                                                       |                                                             |

**Access:** [Help](#) | [Register and Manage License Key\(s\)](#) > [License Management: Add Key](#) > [Add Registration Key](#)

### How to:

To add a private key, follow the steps below:

| Step | Action                                                                             | See also                                           |
|------|------------------------------------------------------------------------------------|----------------------------------------------------|
| 1    | Click on the Enter Private Key tab                                                 | <a href="#">License Management</a> <sup>2094</sup> |
| 2    | In the <b>Name</b> and <b>Company</b> fields, type your user name and company name |                                                    |

| Step | Action                                                                                                                              | See also |
|------|-------------------------------------------------------------------------------------------------------------------------------------|----------|
| 3    | In the <b>Copy registration key into space below...</b> field, copy the registration key (including any parentheses around the key) |          |
| 4    | Click on the <b>OK</b> button to confirm the key selection                                                                          |          |

To add a shared key, follow the steps below:

| Step | Action                                                                                                                                                                                                                                               | See also                                     |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| 1    | Click on the Get Shared Key tab                                                                                                                                                                                                                      | <a href="#">License Management</a><br>[2097] |
| 2    | In the <b>Name</b> and <b>Company</b> fields, type your user name and company name                                                                                                                                                                   |                                              |
| 3    | In the <b>Shared Keystore</b> field, click on the ( ... ) (Browse) button<br>The Shared Keystore Selection dialog displays                                                                                                                           |                                              |
| 4    | If your keystore is file-based: <ul style="list-style-type: none"> <li>• Select the <b>File Based Keystore</b> radio button</li> <li>• Click on the <b>Browse</b> button, and</li> <li>• Locate and select the keystore file</li> </ul> Go to step 7 |                                              |
| 5    | If the keystore is network-based: <ul style="list-style-type: none"> <li>• Click on the <b>Sparx Keystore Server</b> radio button</li> <li>• In the <b>Server Address</b> field, type the server address of the keystore</li> </ul>                  |                                              |
| 6    | If necessary, type in the password (advised by your administrator) and/or click on the <b>Test</b> button to ensure that you have a connection to the keystore                                                                                       |                                              |
| 7    | Click on the <b>OK</b> button to return to the Get Shared Key tab<br>This now shows the name of the keystore in the <b>Shared Keystore</b> field                                                                                                     |                                              |
| 8    | In the <b>Select a Product</b> field, click on the appropriate product name                                                                                                                                                                          |                                              |
| 9    | Click on the <b>OK</b> button<br>The License Management dialog redisplay, indicating that the shared key is registered for the selected product until the key expiry date                                                                            |                                              |
| 10   | Click on the <b>Close</b> button                                                                                                                                                                                                                     |                                              |

## 20.3 Keystore Troubleshooting

| Message Displayed:                                                        | Explanation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><i>Error reading Key Store file:<br/>(Access is denied)</i></p>        | <p>All users who are to use the shared key facility require <b>Read, Write and Modify</b> access to the <i>sskeys.dat</i> file containing the shared keys</p> <p>Please verify that all required users have sufficient permissions to the file and try again</p> <p>Review the effective permissions calculated at the location of the key file for the user account reporting the problem - you should closely examine the permissions for both the Network Share and the File System; it is possible that these permissions have been overwritten at some point</p> <p>If the problem continues, contact Sparx Support</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <p><i>Error reading Key Store file:<br/>(Key File has been moved)</i></p> | <p>In the key store, as a security measure the hard drive serial number is recorded when the file is created; the file then cannot be moved from the original location in which it was created</p> <p>If the key store has to be re-located for any reason, the administrator should re-create the key store in the new location using the original license keys</p> <p>This issue is commonly seen after a file server has undergone a hardware upgrade in which the physical hard drives have been replaced; problems could also occur if the drive used is part of a RAID configuration</p> <p>The message can also appear where the key store exists on a Novell-based file system - when creating the key store, the administrator is prompted to confirm that the key store is to be located on a Novell Netware file server; if they click on the <b>Yes</b> button, the key store instead records the logical path used to create it, and all users must connect to the key store using this same path</p> <p>The recorded path is case-sensitive and must be an exact match</p> |

## 20.4 Upgrade an Existing License

This topic explains how to upgrade your installation of Enterprise Architect to a more powerful edition.

| Topic                          | Detail                                                                                                                                                                                                                                                                                                                                                                                                                    | See also                                                                                                                                                                    |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Editions</b>                | <p>There are six Enterprise Architect editions:</p> <ul style="list-style-type: none"> <li>• Desktop</li> <li>• Professional</li> <li>• Corporate</li> <li>• Business and Software Engineering</li> <li>• Systems Engineering</li> <li>• Ultimate</li> </ul> <p>The installation file provides the facilities of all editions; your registration key gives access to the facilities of the edition you have purchased</p> | <a href="#">Editions Available</a> <sup>[14]</sup>                                                                                                                          |
| <b>Upgrade</b>                 | <p>If you have purchased a less powerful edition, such as Professional or Desktop, you can subsequently upgrade your existing license to a more powerful edition by purchasing a special upgrade key from Sparx Systems</p> <p>Once you have purchased and received the appropriate key, use the procedure below to gain access to the additional features</p>                                                            | <a href="#">Enterprise Architect Purchase Details</a>                                                                                                                       |
| <b>Trial and Lite Editions</b> | <p>The Lite edition and the Trial version cannot be registered or upgraded</p> <p>If you have used either of these versions and then purchased Enterprise Architect, you must download the registered version from:</p> <p style="text-align: center;"><i>www.sparxsystems.com/securedownloads/easetupfull.exe</i></p> <p>Then you can enter your registration key</p>                                                    | <a href="#">The Trial Version</a> <sup>[13]</sup><br><a href="#">The Read-only 'Lite' Edition</a> <sup>[18]</sup><br><a href="#">Registered User software download site</a> |
| <b>Application</b>             | <p>This topic is mainly applicable to users with private keys</p> <p>If you are an end-user with a shared key, you would simply be allocated the relevant key next time you requested one</p> <p>If, however, you want to upgrade while using a shared key on a long lease, you would simply click on the <b>Release Key</b> button and then the <b>Add Key</b> button</p>                                                | <a href="#">Adding License Keys</a> <sup>[2096]</sup>                                                                                                                       |

**Access:** [Help | Register and Manage License Key\(s\)](#) > **License Management: Add Key**

### How to:

To upgrade from one license edition to another, follow the steps below:

| Step | Action                                                                                                                    | See also                                             |
|------|---------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| 1    | Make sure you have a valid upgrade key purchased from Sparx Systems; you typically receive this in an email or PDF format |                                                      |
| 2    | On the Add Registration Key dialog, in the <b>Name</b> and <b>Company</b> fields, type your name and company name         | <a href="#">License Management</a> <sup>[2094]</sup> |

| Step | Action                                                                                                                                                                                                                                                                                                                                  | See also |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 3    | Copy the key you received for the upgraded edition of Enterprise Architect - including the { and } bracket characters - from the email                                                                                                                                                                                                  |          |
| 4    | Paste the key into the <b>Copy registration key</b> field                                                                                                                                                                                                                                                                               |          |
| 5    | Click on the <b>OK</b> button<br>Enterprise Architect displays the message:<br><i>Registration succeeded – Thank you for purchasing Enterprise Architect<br/>xxxx Edition</i>                                                                                                                                                           |          |
| 6    | Click on the <b>OK</b> button, and then on the <b>Close</b> button to continue working in Enterprise Architect                                                                                                                                                                                                                          |          |
| 7    | Select the <b>Help   About EA</b> menu option<br>Copy the registration key shown and store it somewhere safe; this is a key to the full license of the edition you have upgraded to<br>If you ever have to reinstall Enterprise Architect, you can register it with this key, so you won't have to go through the upgrade process again |          |

## 20.5 Register Add-In

Apart from registering Enterprise Architect, you must also register any Add-Ins you purchase to use with the application

This topic explains how to register a private key; if you have purchased a floating licence (shared key) for the Add-In, follow the *Add a Shared Key* section of the *Adding License Keys* topic.

**Access:** [Help | Register and Manage License Key\(s\) Add-Ins | Enter License Key for <Add-In name>](#)

### How To:

To register Add-Ins for Enterprise Architect, follow the steps below:

| Step | Action                                                                                                                                                                                                                                                                 | See Also                                                  |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| 1    | Purchase one or more licenses for the Add-In from your Add-In provider<br>Once you have paid for a licensed version of the Add-In, you receive (via email or other suitable means) a license key for the product                                                       |                                                           |
| 2    | Save the license key and the latest full version of the Add-In                                                                                                                                                                                                         |                                                           |
| 3    | Run the Add-In's setup program to install the Add-In                                                                                                                                                                                                                   |                                                           |
| 4    | In Enterprise Architect, select either of the menu options: <ul style="list-style-type: none"> <li>• <b>Register and Manage License Key(s)</b></li> <li>• <b>Add-Ins   Enter License Key for &lt;Add-In name&gt;</b></li> </ul> The License Management dialog displays | <a href="#">License Management</a><br>t <sup>[2094]</sup> |
| 5    | Click on the <b>Add Key</b> button<br>The Add Registration Key dialog displays, showing the Enter Private Key tab                                                                                                                                                      |                                                           |
| 6    | Type or copy-and-paste in the key you received with the Add-In, including the { and } characters                                                                                                                                                                       |                                                           |
| 7    | Click on the <b>OK</b> button                                                                                                                                                                                                                                          |                                                           |
| 8    | When the Add-In has been added successfully, close down Enterprise Architect and restart it to apply the integration changes imposed by the Add-In                                                                                                                     |                                                           |

### Learn More:

- [Adding License Keys](#)  
t<sup>[2096]</sup>

# Index

■ ■ ■

## .EMX

Import 323

## .NET

ASP, Debug 1431  
 Debug 1428  
 Debug Another Process 1448  
 Debug CLR Versions 1429  
 Debug With COM Interop Process 1430  
 Debug, System Requirements 1417  
 Garbage Collect (Exit Method) 1855  
 Set Up Debug Session 1429

## .UML2

Import 323

## - A -

### Abstract

Complex Models 954  
 XSD Models 1595

Acceptance Testing 1700

### AcceptEvent Action

Triggers Tab 866

### Access

Email 128  
 Internet Search Engine 128  
 MDG Technologies Remote From Enterprise Architect 1034  
 Web Site 128

### Access 2007

Repository, Create 149  
 Upsize To 174

### Acknowledgement

CXImage Library 25  
 Of Contributions 25  
 Of Trademarks 23  
 Print Listview 25

### Action

AcceptEvent, Triggers Tab 866  
 As Hyperlink 1294  
 BroadcastSignal, Signal Tab 866  
 Call 1504  
 Element 865  
 Expansion Node 868  
 Invocation 1504  
 Local Pre/Post Conditions 871

Notation 866  
 Operations 865, 872  
 SendSignal, Signal Tab 866  
 StructuralFeature 866  
 Trigger, AcceptEvent 866  
 Type, Set 866  
 Update Operation 865, 872

### Action Pin

Add To Action 869  
 As Action Property 870  
 As Argument For Call Action 869  
 Assign To Action 870  
 Properties 869

### ActionScript

Code Generation Language Options 1530  
 Import, Reverse Engineering 1510  
 Modeling Conventions 1458  
 Options 1530  
 Versions Supported 1530

### Activate

MDG Technologies 1033

Activate Recording Markers 1647

### Activation

End 855  
 Extend Down 855  
 Extend Up 855  
 Lower 855  
 Raise 855  
 Sequence Element 855  
 Suppress 855

### Activation Layer

Sequence Diagram Lifelines 856

### Activation Levels

Sequence Diagram Lifeline Self Messages 856

### Active

Classes 943  
 State Configuration 919

### Active Directory

Import User ID From 199

### Active State Logic

Model State Machine For HDL 1502

### Activity

Behavioral Aspects 715  
 Edit Parameters 715  
 Element 873  
 Elements And Connectors 558  
 Group, Toolbox 558  
 Instance 1013  
 IsException 875  
 IsStream 875  
 Notation 874  
 Parameter Nodes 875

- Activity
  - Parameters, Define 717
  - Partition 876, 914
  - Paste As Action 579
  - Paste As Link 579
  - Paste From Project Browser 579
  - Process Element 1297
  - Region Element 916
  - Structured 927
  - Structured, Conditional Node 926, 929
  - Structured, Loop Node 926, 928
  - Structured, Sequential 926
- Activity (BPEL)
  - Create 1244
  - Loop 1244
  - Model 1244
  - Sub-Process 1244
  - Task 1244
  - Types 1244
- Activity Diagram
  - Create Object From Attribute 704
  - Description 811
  - Elements And Connectors 811
  - Example 813
  - Generate Code From 1496, 1504
  - Generate From Scenario 676, 677
  - Generate Scenario From 684
  - Object Flows 1004
  - Operations 865, 872
- Activity Edge
  - Connector 985
  - Relationship 985
- Activity Final
  - Element 895
- Activity Partition
  - Docking 914
  - Element 914
  - Horizontal 914
  - Vertical 914
- Actor
  - Element 877
- Ada 2005
  - Code Generation Language Options 1531
  - Modeling Conventions 1459
  - Options 1531
- Adaptive Server Anywhere
  - Data Repository, Connect To 169
  - ODBC Driver, Set Up 158
  - Repository, Create 152
  - Upsize To 180
- Add
  - Category To Team Review 220
  - Code Modules In MDG Technology Wizard 1070
  - Connectors Between Locked Elements 211
  - Connectors To UML Model, Quick Start 42
  - Diagram Properties Note 587
  - Diagram To Package 40
  - Diagram to Project 568
  - Diagram To UML Model, Quick Start 40
  - Diagram Type In MDG Technology Wizard 1068
  - Element Changes 1715
  - Element Defects 1715
  - Element Directly To Package 627
  - Element Issues 1715
  - Element Tasks 1715
  - Element To Diagram 41
  - Element To Diagram From Project Browser 576
  - Element To UML Model, Quick Start 41
  - Enumeration Tags To Stereotypes 1047
  - Filters To Search 487
  - Images In MDG Technology Wizard 1071
  - Instance Variable 958
  - Key 2096
  - Learning Center In MDG Technology Wizard 1069
  - License Key 2096
  - Line Points 745
  - Linked Document Template In MDG Technology Wizard 1072
  - MDA Transforms In MDG Technology Wizard 1071
  - Model To Project 136
  - MS Word Table Of Contents 1798
  - MS Word Table Of Figures 1799
  - New Code Sections To Existing Features 1488
  - Note To Connector 743
  - Note To Link 743
  - Package In Project Browser 534
  - Package To UML Model, Quick Start 39
  - Packages To Model Document 1778
  - Pattern In MDG Technology Wizard 1067
  - Pattern To Diagram 1024, 1025
  - Port To Element 962
  - Post To Team Review 222
  - Profile Attribute To Diagram 1028
  - Profile Connector To Diagram 1028
  - Profile Element To Diagram 1028
  - Profile In MDG Technology Wizard 1066
  - Profile Operation To Diagram 1028
  - Project Items Via Toolbar 108
  - Project Task 357
  - Property Value to Part 961

- Add
  - RTF Report Template In MDG Technology Wizard 1072
  - Scripts In MDG Technology Wizard 1072
  - Shape Script To Stereotype In Profile 1048
  - Tagged Value Types In MDG Technology Wizard 1069
  - Tagged Values 763
  - Test Details 1696
  - Toolbox In MDG Technology Wizard 1068
  - Topic To Team Review 221
  - UML Diagram 568
  - UML Pattern To Diagram 1024, 1025
  - Views 531
- Add And Delete Attributes
  - Automation Interface Code Example 1957
- Add And Delete Methods
  - Automation Interface Code Example 1957
- Add And Manage Diagrams
  - Automation Interface Code Example 1957
- Add And Manage Elements
  - Automation Interface Code Example 1955
- Add And Manage Packages
  - Automation Interface Code Example 1954
- Add Connector
  - Automation Interface Code Example 1956
- Add Stereotypes
  - Automation Interface Code Example 1963
- Add Submenu
  - Element Context Menu, Project Browser 450
  - Package Context Menu, Project Browser 446
- Add to Project Clipboard
  - Menu Option (Edit Menu) 74
- Add-In
  - And Enterprise Architect Deadlocks (.NET) 1972
  - COM Interoperability 1972
  - Concurrent Method Calls 1972
  - Connect To 97
  - Create 1968
  - Create, Define Menu Items 1969
  - Deploy 1970
  - Disable 1974
  - Display Help On 97
  - Enable 1974
  - Events 1976
  - Holding State Information 1972
  - Manage 1974
  - Manager 1974
  - MDG 1036
  - Menu 97
  - Model Driven Generation 1036
  - Pre-2004 1972
  - Re-entrancy 1972
  - Register 2101
  - Run Functions From Learning Center 1084
  - Search 484, 1974
  - Search Data 1975
  - Submenu 97
  - Tasks 1968
  - Visual Basic Issues 1972
- Add-In Event
  - EA\_Connect 1977
  - EA\_Disconnect 1977
  - EA\_GetMenuItems 1978
  - EA\_GetmenuState 1979
  - EA\_MenuClick 1980
  - EA\_OnOutputItemClicked 1981
  - EA\_OnOutputItemDoubleClicked 1982
  - EA\_ShowHelp 1983
- Add-In License Management Event
  - EA\_AddinLicenseGetDescription 1986
  - EA\_AddinLicenseValidate 1985
- Add-In License Management Events
  - Add-In Model 1984
- Add-In Model
  - Add-In Event, EA\_Connect 1977
  - Add-In Event, EA\_Disconnect 1977
  - Add-In Event, EA\_GetMenuItems 1978
  - Add-In Event, EA\_GetMenuState 1979
  - Add-In Event, EA\_MenuClick 1980
  - Add-In Event, EA\_OnOutputItemClicked 1981
  - Add-In Event, EA\_OnOutputItemDoubleClicked 1982
  - Add-In Event, EA\_ShowHelp 1983
  - Add-In Event, Overview 1976
  - Add-In License Management Event, EA\_AddinLicenseGetDescription 1986
  - Add-In License Management Event, EA\_AddinLicenseValidate 1985
  - Add-In License Management Events 1984
  - Add-In Tasks 1968
  - Benefits 1967
  - Broadcast Event, EA\_FileClose 1993
  - Broadcast Event, EA\_FileNew 1993
  - Broadcast Event, EA\_FileOpen 1994
  - Broadcast Event, EA\_OnPostCloseDiagram 1995
  - Broadcast Event, EA\_OnPostInitialized 1995
  - Broadcast Event, EA\_OnPostOpenDiagram 1996
  - Broadcast Event, EA\_OnPostTransform 1997
  - Broadcast Event, EA\_OnPreExitInstance 1997

- Add-In Model
- Broadcast Event, EA\_OnRetrieveModelTemplate 1998
  - Broadcast Events 1983
  - Compartment Events 1986
  - Compartment Events, EA\_GetCompartmentData 1987
  - Compartment Events, EA\_QueryAvailableCompartments 1986
  - Context Item Events 1989
  - Context Item Events, EA\_OnContextItemChanged 1990, 1992
  - Context Item Events, EA\_OnContextItemDoubleClicked 1991
  - Create Add-In 1968
  - Create Add-In, Tricks and Traps 1972
  - Create Custom View 2041
  - Custom View 2040
  - EA\_AddinLicenseGetDescription 1986
  - EA\_AddinLicenseValidate 1985
  - EA\_Connect 1977
  - EA\_Disconnect 1977
  - EA\_FileClose 1993
  - EA\_FileNew 1993
  - EA\_FileOpen 1994
  - EA\_GetCompartmentData 1987
  - EA\_GetMenuItems 1978
  - EA\_GetMenuState 1979
  - EA\_MenuClick 1980
  - EA\_OnAttributeTagEdit 2031
  - EA\_OnConnectorTagEdit 2032
  - EA\_OnContextItemChanged 1990, 1992
  - EA\_OnContextItemDoubleClicked 1991
  - EA\_OnDeleteTechnology 2038
  - EA\_OnElementTagEdit 2033
  - EA\_OnEndValidation 2001
  - EA\_OnImportTechnology 2039
  - EA\_OnInitializeTechnologies 2035
  - EA\_OnInitializeUserRules 2000
  - EA\_OnOperationTagEdit 2034
  - EA\_OnOutputItemClicked 1981
  - EA\_OnOutputItemDoubleClicked 1982
  - EA\_OnPostActivateTechnology 2037
  - EA\_OnPostCloseDiagram 1995
  - EA\_OnPostInitialized 1995
  - EA\_OnPostNewAttribute 2014
  - EA\_OnPostNewConnector 2011
  - EA\_OnPostNewDiagram 2012
  - EA\_OnPostNewDiagramObject 2013
  - EA\_OnPostNewElement 2011
  - EA\_OnPostNewGlossaryTerm 2016
  - EA\_OnPostNewMethod 2015
  - EA\_OnPostNewPackage 2015
  - EA\_OnPostOpenDiagram 1996
  - EA\_OnPostTransform 1997
  - EA\_OnPreActivateTechnology 2036
  - EA\_OnPreDeleteAttribute 2018
  - EA\_OnPreDeleteConnector 2020
  - EA\_OnPreDeleteDiagram 2020
  - EA\_OnPreDeleteDiagramObject 2021
  - EA\_OnPreDeleteElement 2017
  - EA\_OnPreDeleteGlossaryTerm 2023
  - EA\_OnPreDeleteMethod 2019
  - EA\_OnPreDeletePackage 2022
  - EA\_OnPreDeleteTechnology 2037
  - EA\_OnPreExitInstance 1997
  - EA\_OnPreNewAttribute 2027
  - EA\_OnPreNewConnector 2025
  - EA\_OnPreNewDiagram 2026
  - EA\_OnPreNewDiagramObject 2026
  - EA\_OnPreNewElement 2024
  - EA\_OnPreNewGlossaryTerm 2030
  - EA\_OnPreNewMethod 2028
  - EA\_OnPreNewPackage 2029
  - EA\_OnRetrieveModelTemplate 1998
  - EA\_OnRunAttributeRule 2004
  - EA\_OnRunConnectorRule 2003
  - EA\_OnRunDiagramRule 2002
  - EA\_OnRunElementRule 2001
  - EA\_OnRunMethodRule 2005
  - EA\_OnRunPackageRule 2002
  - EA\_OnRunParameterRule 2005
  - EA\_OnStartValidation 2000
  - EA\_QueryAvailableCompartments 1986
  - EA\_ShowHelp 1983
  - Interface 1967
  - Introduction 1967
  - MDG Add-Ins 2042
  - MDG Add-Ins, MDG Events 2042
  - MDG Events, MDG\_BuildProject 2043
  - MDG Events, MDG\_Connect 2044
  - MDG Events, MDG\_Disconnect 2045
  - MDG Events, MDG\_GetConnectedPackages 2046
  - MDG Events, MDG\_GetProperty 2046
  - MDG Events, MDG\_Merge 2047
  - MDG Events, MDG\_NewClass 2049
  - MDG Events, MDG\_PostGenerate 2050
  - MDG Events, MDG\_PostMerge 2051
  - MDG Events, MDG\_PreGenerate 2051
  - MDG Events, MDG\_PreMerge 2052
  - MDG Events, MDG\_PreReverse 2053
  - MDG Events, MDG\_RunExe 2054
  - MDG Events, MDG\_View 2054

- Add-In Model
  - Model Validation Broadcasts 1999
  - Model Validation Broadcasts, EA\_OnEndValidation 2001
  - Model Validation Broadcasts, EA\_OnInitializeUserRules 2000
  - Model Validation Broadcasts, EA\_OnRunAttributeRule 2004
  - Model Validation Broadcasts, EA\_OnRunConnectorRule 2003
  - Model Validation Broadcasts, EA\_OnRunDiagramRule 2002
  - Model Validation Broadcasts, EA\_OnRunElementRule 2001
  - Model Validation Broadcasts, EA\_OnRunMethodRule 2005
  - Model Validation Broadcasts, EA\_OnRunPackageRule 2002
  - Model Validation Broadcasts, EA\_OnRunParameterRule 2005
  - Model Validation Broadcasts, EA\_OnStartValidation 2000
  - Model Validation Example 2006
  - Post-New Events 2010
  - Post-New Events, EA\_OnPostNewAttribute 2014
  - Post-New Events, EA\_OnPostNewConnector 2011
  - Post-New Events, EA\_OnPostNewDiagram 2012
  - Post-New Events, EA\_OnPostNewDiagramObject 2013
  - Post-New Events, EA\_OnPostNewElement 2011
  - Post-New Events, EA\_OnPostNewGlossaryTerm 2016
  - Post-New Events, EA\_OnPostNewMethod 2015
  - Post-New Events, EA\_OnPostNewPackage 2015
  - Pre-Deletion Events 2017
  - Pre-Deletion Events, EA\_OnPreDeleteAttribute 2018
  - Pre-Deletion Events, EA\_OnPreDeleteConnector 2020
  - Pre-Deletion Events, EA\_OnPreDeleteDiagram 2020
  - Pre-Deletion Events, EA\_OnPreDeleteDiagramObject 2021
  - Pre-Deletion Events, EA\_OnPreDeleteElement 2017
  - Pre-Deletion Events, EA\_OnPreDeleteGlossaryTerm 2023
  - Pre-Deletion Events, EA\_OnPreDeleteMethod 2019
  - Pre-Deletion Events, EA\_OnPreDeletePackage 2022
  - Pre-New Events 2023
  - Pre-New Events, EA\_OnPreNewAttribute 2027
  - Pre-New Events, EA\_OnPreNewConnector 2025
  - Pre-New Events, EA\_OnPreNewDiagram 2026
  - Pre-New Events, EA\_OnPreNewDiagramObject 2026
  - Pre-New Events, EA\_OnPreNewElement 2024
  - Pre-New Events, EA\_OnPreNewGlossaryTerm 2030
  - Pre-New Events, EA\_OnPreNewMethod 2028
  - Pre-New Events, EA\_OnPreNewPackage 2029
  - Search Data, XML Format 1975
  - Tagged Value Broadcast Events 2031
  - Tagged Value Broadcast Events, EA\_OnAttributeTagEdit 2031
  - Tagged Value Broadcast Events, EA\_OnConnectorTagEdit 2032
  - Tagged Value Broadcast Events, EA\_OnElementTagEdit 2033
  - Tagged Value Broadcast Events, EA\_OnOperationTagEdit 2034
  - Technology Event, EA\_OnInitializeTechnologies 2035
  - Technology Events 2034
  - Technology Events, EA\_OnDeleteTechnology 2038
  - Technology Events, EA\_OnImportTechnology 2039
  - Technology Events, EA\_OnPostActivateTechnology 2037
  - Technology Events, EA\_OnPreActivateTechnology 2036
  - Technology Events, EA\_OnPreDeleteTechnology 2037
- Additional Collection Class 1528
- Administrator
  - Security Permissions 196
- Advanced
  - Settings, Connector 761
  - Settings, Elements 664
  - Tag Management 767
- Advanced (Element)
  - Submenu 86
- Advanced Options
  - RTF Report 1767
- Advanced Search Options 486
- Aggregate
  - Connector 968
  - Relationship 968
- Aggregation Connector

- Aggregation Connector
  - Change Form 969
- Align
  - Elements From Toolbar 111
  - Multiple Elements 635
- Alignment
  - Submenu 88
- All User Permissions
  - Dialog, User Security 202
  - View, User Security 202
- Allocated Resources
  - View, Project Calendar 396
- Allocated Work
  - Tab, Personal Information Window 382
- Alternate Path
  - Add To Scenario (Toolbar) 671
  - RTF Report Template Section 1741
- Alternate Path, Scenario 667
- Alternative Image
  - For Element 593
  - In Diagram 593
  - Select 593
  - Stereotype 1020
- Analysis
  - Elements and Connectors 561
  - Group, Toolbox 561
  - Models, Business Processes 1195
  - Stereotypes 1186
- Analysis Diagram
  - Description 1187
  - Diagram 1187
  - Elements And Connectors 1187
  - Example 1189
- Analyzer
  - Menu 94
- Analyzer Script 1663
  - Options, Filters 1639
- Analyzer Scripts
  - Manage, Visual Execution Analyzer 1392
- and any breakpoints should 1431
- ANSI C 1531
  - Modeling Conventions 1461
- Anti-Aliased Rendering
  - Of Diagrams 427
- Anti-Aliased Text
  - In Diagrams 427
- Apache Tomcat
  - Server Configuration 1428
  - Server, Debugging 1425
  - Service Configuration 1428
- App Object
  - Automation Interface 1830
- APPDATA
  - Import MDG Technologies 1035
- Appearance
  - Apply From Clipboard 655
  - Connectors Context Menu Section 740
  - Copy For Element 655
  - Default, Of Element 641
  - Element Context Menu 655
- Appearance (Element)
  - Autosize 88
  - Submenu 88
- Applets
  - Java, In Internet Browsers, Debug 1424
- Application
  - Connector 560
- Application Look Dialog 125
- Application Workspace 69
- Apply
  - Rigorous Security Mode Lock 213
  - RTF Report Filter (Legacy) 1789
  - Stereotype To Dependency Relationship 979
  - Stereotype To Element 1017
  - Stereotype To UML Construct 1017
  - User Lock 213
- ArchiMate
  - Concept 1272
  - Diagram 1272
  - Disable 1272
  - Elements 1272
  - Enable 1272
  - MDG Technology 1272
  - Relationships 1272
  - Toolbox Pages 1272
- Argument
  - Call, Synchronize With Behavior Parameter 716
  - For Behavioral Parameter 716
  - Invocation, Synchronize With Behavior Parameter 716
- Arrange
  - Connectors 744
- Array
  - View Elements Of, Debugger 1443
- Arrow
  - Quick Linker 622
- Artifact
  - Element 940, 941
- ASA
  - Data Repository, Connect To 169
  - ODBC Driver, Set Up 158
  - Repository, Create 152
  - Upsize To 180
- ASP .NET

- ASP .NET
  - Debug 1431
- Assembly
  - Connector 969
  - Relationship 969
- Assign
  - Information to Tagged Values 765
  - People To Changes 1722
  - People To Defects 1722
  - Tagged Values To Item 764
- Assignments
  - BPEL, Create 1250
- Associate
  - Calls With Behaviors 716
  - Connector 970
  - Invocations With Behaviors 716
  - Relationship 970
- Associated Files
  - Elements 687
- Association 756
  - Class 1296
  - Connector 970
  - Connector, Set Collection Class 1526
  - Details 756
  - Dot On Connector 758
  - N-Ary 1296
  - Properties 756
  - Relationship 970
  - Set Derived Property 738
  - Specialisation, Set Up 753
- Association Class
  - Connector 973
  - Find In Project Browser 737
  - Link New Class To Association 975
  - Locate In Project Browser 973
  - Relationship 973
  - Shape Scripts 1048
- Association End
  - Qualifiers 971, 972
- Asynchronous Signal Message
  - Associate With Signal 996
  - Connector 996
  - Relationship 996
- Attach
  - Note To Link 743
- Attach To Process Dialog 1448
- Attribute 454
  - Add And Delete, Automation Interface Code Example 1957
  - Add To Element, In-place Editor 726
  - Automatic Properties, C# 701
  - Collections 698
  - Connect To 742
  - Constraints 699
  - Context Menu, Project Browser 452
  - Copy Between Elements 646
  - Create 695
  - Create Fast, Option 1522
  - Create Object From 704
  - Definition 695
  - Delete 696
  - Delete If Not In Code In Reverse Synchronization 1522
  - Derived 696
  - Dialog 695
  - Dialog, Constraints Tab 699
  - Dialog, Detail Tab 698
  - Dialog, General Tab 696
  - Disconnect From 742
  - Edit Keyword 724
  - Edit Name, In-Place Editor 722
  - Edit Scope 723
  - Edit Stereotype, In-place Editor 722
  - Fast Create 695
  - Imported, Default Name Generated From 1522
  - Inherited, Display 702
  - Inherited, Show 585
  - Introduction 695
  - Link To Element Via Object 704
  - Message Part, WSDL 1614
  - Modify 695
  - Move Between Elements 647
  - Multiplicity 698
  - Of Toolbox Page 1075
  - PData & StyleEx, Diagram Profiles 1081
  - Private, Icon 454
  - Properties, Create 700
  - Protected, Icon 454
  - Qualifiers 698, 972
  - Redefine Property 698
  - Show On Diagram 585
  - Stereotyped, For Columns 1351
  - Subsetted Property 698
  - Supported, Create Composite Elements 1056
  - Supported, Define Child Diagram Types 1057
  - Supported, Metatype In Profiles 1051
  - Supported, Stereotype In Profiles 1051
  - Tagged Values 699
  - UML Property, isUnique 698
  - Work With, Automation Interface Code Example 1964
- Attribute Class
  - Alias, Automation Interface 1898

- Attribute Class
  - Automation Interface, ElementFeatures Package 1898
- AttributeConstraint
  - Automation Interface, ElementFeatures Package 1900
- AttributeTag
  - Automation Interface, ElementFeatures Package 1901
- Audit
  - Requirements 1171
  - Scope 300
- Audit History Tab
  - Description 307
  - How To Display 307
  - On Output Window 127, 459
- Audit Options
  - All 300
  - Connectors Audited 300
  - Core Structural 300
  - Custom 300
  - Elements Audited 300
  - Maintenance 300
- Audit View
  - Advanced Mode 304
  - Audit Changes 304
  - Controls 304
  - Custom Time Periods 304
  - Deleted Mode 304
  - Display Database Changes 304
  - Filter By Time 304
  - Performance Problems 308
  - Raw Mode 304
  - Refresh 304
  - Search 304
  - Slow Loading 308
  - Slow Navigation 308
  - Sort 304
  - Standard Mode 304
- Auditing
  - Alternative To Differencing 308
  - And Performance Of Enterprise Architect 298
  - And RTF Reporting 298
  - Audit Tree 302
  - Clear Logs 300
  - Display Audit Results 302
  - Enable 300
  - How To Invoke 300
  - Include Reverse Engineering 300
  - Include XMI Export 300
  - Include XMI Import 300
  - Introduction 298
  - Large Deletion Issue 307
  - Level, Core 300
  - Level, Extended 300
  - Level, Standard 300
  - Load Logs 300
  - Performance Issues 307
  - Quick Start 300
  - Record Display 302
  - Reverse Engineering Issue 307
  - Save Logs 300
  - Settings 300
  - Use Database Timestamp 300
  - View 302
  - XMI Import Issue 307
- Authentication
  - Accept Windows Authentication 198
  - Automatic Delete On Relocation Of Project 198
- Author
  - Attributes 1839
  - Define 778
  - From Windows Active Directory 778
  - Methods 1839
- Author Collection
  - Automation Interface Repository 1839
- Auto Counter
  - Of Elements 628
- Auto Increment Column Values
  - Impact Of Replication 416
  - Impact Of XMI Import 416
  - Reset 416
- Auto Naming
  - Of Element 628
- Auto Numbering
  - Of Elements 628
- Auto Route Diagram Layout 618
- Auto Save
  - Diagram Changes 430
- Autocompletion List
  - Code Editor, Common 1399
- Autohide
  - Reveal Autohidden Window 105
  - Turn Off 105
  - Turn On 105
  - Windows 105
- Autolayout
  - Diagram 619
  - Options 619
- Automatic
  - Save Of Diagram Changes 430
- Automatic Exclusive Edit Lock 196
- Automatic Indentation
  - Code Editor, Common 1397

- Automatically Hidden Windows
  - Animate 78
- Automation
  - Of Enterprise Architect 1816
- Automation Interface 1909
  - App Object 1830
  - Attribute Class, ElementFeatures Package 1898
  - AttributeConstraint, ElementFeatures Package 1900
  - AttributeTag, ElementFeatures Package 1901
  - Available Resources 1828
  - Call Executables From Enterprise Architect 1822
  - Call From Enterprise Architect 1827
  - Code Example, Add And Delete Attributes 1957
  - Code Example, Add And Delete Methods 1957
  - Code Example, Add And Manage Diagrams 1957
  - Code Example, Add And Manage Elements 1955
  - Code Example, Add And Manage Packages 1954
  - Code Example, Add Connector 1956
  - Code Example, Add Stereotypes 1963
  - Code Example, Iterate Through EAP File 1953
  - Code Example, Open The Repository 1953
  - Code Example, Use Element Extras 1958
  - Code Example, Use Repository Extras 1961
  - Code Example, Work With Attributes 1964
  - Code Example, Work With Methods 1965
  - Code Examples, Introduction 1952
  - Connect From Borland Delphi 7.0 1823
  - Connect From Java 1823
  - Connect From MS C# 1823
  - Connect From MS Visual Basic 6.0 1823
  - Connect To 1822
  - Connector Package Diagram 1910
  - Connector, Connector Package 1911
  - ConnectorConstraint, Connector Package 1915
  - ConnectorEnd, Connector Package 1916
  - ConnectorTag, Connector Package 1918
  - ConstLayoutStyles Enum 1832
  - Constraint, Element Package 1876
  - CreateBaselineFlag Enum 1832
  - CreateModelType Enum 1832
  - CustomProperties Collection, ElementFeatures Package 1902
  - Diagram Package 1920
  - Diagram, Diagram Package 1921
  - DiagramLinks, Diagram Package 1924
  - DiagramObject, Diagram Package 1925
  - Document Generator Interface 1947
  - DocumentBreak Enum 1833
  - DocumentGeneratorClass 1947
  - DocumentType Enum 1833
  - EAEditionTypes Enum 1833
  - Effort, Element Package 1876
  - Element Package Diagram 1873
  - Element Package, File 1887
  - Element, Element Package 1877
  - ElementFeatures Package Diagram 1897
  - EmbeddedElements Collection, ElementFeatures Package 1902
  - Enumerations 1831
  - EnumRelationSetType Enum 1834
  - Examples 1822
  - Examples and Tips 1826
  - ExportPackageXMIFlag Enum 1834
  - Introduction 1822
  - Issue, Element Package 1888
  - Mail Interface Package 1950
  - MailInterface Class, Mail Interface Package 1950
  - MDGMenus Enum 1834
  - MessageFlag 1835
  - Method, ElementFeatures Package 1903
  - MethodConstraint, ElementFeatures Package 1905
  - MethodTag, ElementFeatures Package 1906
  - Metric, Element Package 1889
  - Model 1829
  - ObjectType Enum 1835
  - Package 1829
  - Parameter, ElementFeatures Package 1906
  - ParamTag, ElementFeatures Package 1908
  - Partitions Collection, ElementFeatures Package 1908
  - Project Interface 1929
  - Project, Project Interface 1929
  - Properties, ElementFeatures Package 1909
  - Property ElementFeatures Package 1909
  - PropType Enum 1836
  - Reference 1828
  - ReloadType Enum 1837
  - Repository 1855
  - Repository Package 1838, 1845
  - Repository, Author Collection 1839
  - Repository, Client Collection 1840
  - Repository, Collection Class 1841
  - Repository, Datatype 1843
  - Repository, EventProperties 1844
  - Repository, EventProperty 1844
  - Repository, ModelWatcher 1845

- Automation Interface 1909
  - Repository, ProjectIssues 1851
  - Repository, ProjectResource 1852
  - Repository, ProjectRole 1853
  - Repository, PropertyType 1853
  - Repository, Reference 1854
  - Repository, Stereotype 1871
  - Repository, Task 1872
  - Repository, Term 1873
  - Requirement, Element Package 1890
  - Resource, Element Package 1891
  - Risk, Element Package 1891
  - RoleTag, Connector Package 1919
  - Scenario, Element Package 1892
  - ScenarioDiagramType Enum 1837
  - ScenarioExtension, Element Package 1893
  - ScenarioStep, Element Package 1894
  - ScenarioStepType Enum 1838
  - ScenarioTestType Enum 1838
  - Set Up Visual Basic 1825
  - Swimlane, Diagram Package 1928
  - SwimlaneDef, Diagram Package 1927
  - Swimlanes, Diagram Package 1927
  - TaggedValue, Element Package 1895
  - Test, Element Package 1896
  - Transitions Collection, ElementFeatures Package 1910
  - Using 1822
  - VB GetObject Support 1830
  - XMIType Enum 1838
- Autonumbering
  - And Requirements 1156
- Autosize
  - Element, Single 589
  - Elements, As Group 589
- Available Resources
  - Automation Interface 1828
- B -**
- Base Project
  - Copy 146
  - New 146
- Baseline
  - And Differences, Overview 308
  - Considerations 310
  - Create 312
  - Delete 311
  - Export 311
  - Import 311
  - Include XMI Stubs 312
  - Load From Alternative Model 311
  - Manage 311
  - Merge With Current Model, Overview 308
  - Model 310
  - Overview 310
  - Requirements 1171
  - Scenarios 310
  - Versions 312
- BaseModel Script
  - InnoDB 172
  - MyISAM 172
- Basic Path, Scenario 667
- Batch Generate
  - Elements With Code 88
  - RTF Resource Documents, Automatically 1766
  - RTF Resource Documents, Manually 1766
- Batch XMI
  - Export 332
  - Import 333
- Behavior
  - Associate With Calls 716
  - Associate With Invocations 716
  - Instance 1013
  - Of Operation, Display In Diagram 710
  - Operation, Associate With 710
  - Parameter, Tagged Values 718
  - Parameters, Define 717
  - Synchronize Call Arguments With Parameters 716
  - Synchronize Invocation Arguments With Parameters 716
- Behavior Call
  - Actions For RuleTask Operations 1209
  - Interaction Occurrence 715
  - Modeling 715
  - Parameter Arguments 716
- Behavioral Diagram
  - Elements 864
  - Overview 811
- Behavioral Model Templates 1137
- Behavioral Modeling 706
- Behavioral Models
  - Generate Code From 1496
- Behavioral Parameter
  - Edit 717
  - Extend 717
  - Reassign 717
  - Set 717
- Behaviors
  - Of Activities 715
  - Of Interactions 715
- Bend
  - Connector At Cursor 745

- Bend Connector 42
- Bezier Lines 745
- Binary Module
  - Import, Reverse Engineering 1513
- Bind
  - Breakpoints 1439
  - Fail To, Breakpoints 1439
- Binding
  - In Connector 758
  - WSDL Diagram 1617
  - WSDL Element 1617
- Binding Class
  - And Binding Connector 1010
- Binding Expression
  - Binding Connector 1010
- Bitmap Image
  - In Diagrams 593
- Blue Exclamation Mark 214
- Bookmark
  - Clear 74
  - Clear All 74
  - Code Editor, Common 1396
  - For RTF Report 1796
  - Insert In RTF Template 1755
  - Multiple Elements 412
  - Package As 1796
  - RTF, In Master Document Element 1774
  - Selected Element 74
  - Triangle 412
- Bookmark Selected
  - Menu Option (Edit Menu) 74
- Border
  - Red 214
- Boundary
  - Element 1288
  - Element Settings 932
  - Element, Create 1289
  - Object Settings 932
  - Properties 932
- Box Diagram Layout 609
- BPEL
  - Assignments, Create 1250
  - BPMN Elements Not Mapable 1240
  - Concepts 1235
  - Create Assignments 1250
  - Create Model 1238
  - Diagram 1235
  - Disable 1235
  - End Event, Model 1247
  - Generate 1253
  - In Enterprise Architect 1235
  - Intermediate Event, Model 1242
  - Looping Constructs 1244
  - MDG Technology 1235
  - Model A Process 1240
  - Model Activity 1244
  - Model Gateway 1246
  - Model Pool 1249
  - Model Sequence Flow Connector 1248
  - Model, Create 1238
  - Modeling Restrictions 1235
  - Package Structure 1238
  - Process Element, BPMN 1.1 1220
  - Process, Properties 1240
  - Start Event, Model 1240
  - Sub Process Types 1244
  - Task Types 1244
  - Web Service, Create 1251
- BPMN
  - 1.0 1219
  - 1.0 Elements 1220
  - 1.0 Relationships 1220
  - 1.0 Toolbox Pages 1220
  - 1.1 1219
  - 1.1 Elements 1220
  - 1.1 Relationships 1220
  - 1.1 Toolbox Pages 1220
  - 2.0 1219
  - 2.0 Business Process Elements 1222
  - 2.0 Business Process Relationships 1222
  - 2.0 Business Process Toolbox Pages 1222
  - 2.0 Choreography Elements 1224
  - 2.0 Choreography Relationships 1224
  - 2.0 Choreography Toolbox Pages 1224
  - 2.0 Conversation Elements 1225
  - 2.0 Conversation Relationships 1225
  - 2.0 Conversation Toolbox Pages 1225
  - 2.0 Elements 1222
  - 2.0 Relationships 1222
  - 2.0 Toolbox Pages 1222
  - 2.0 Type Elements 1227
  - 2.0 Type Relationships 1227
  - 2.0 Type Toolbox Pages 1227
  - Change Element Appearance 1228
  - Concepts 1219
  - Core Page 1220
  - Diagram 1219
  - Disable 1219
  - Element Appearance, Change 1228
  - MDG Technology 1219
  - Migrate 1.0 Model To 1.1 1232
  - Migrate 1.1 Model To 2.0 1233
  - Toolbox Page 1219
  - Types Page 1220

- BPMN
  - Update Version Via Toolbox Element 1228
  - Version Differences In Appearance 1228
- BPMN 1.1
  - Activity Types 1244
  - BPEL Sequence Flow 1248
  - Elements Not Mappable to BPEL 1240
  - End Event, Types 1247
  - Gateway Types 1246
  - Intermediate Event, Types 1242
  - Pool 1249
  - Start Event, Types 1240
  - Sub-Process Types 1244
  - Task Types 1244
- Brace Matching
  - Code Editor, Common 1397
- Branching Macros
  - Code Template Syntax 1133
- Breakpoint
  - Delete 1436, 1439
  - Difference From Recording Marker 1642
  - Disable 1436, 1439
  - Enable 1436, 1439
  - Fail To Bind 1439
  - Failure Message 1436
  - Management 1436
  - Model Simulation, Using 1684
  - Properties 1437
  - Set For Modifiable Data 1438
  - Set In Code 1438
  - Sets, Model Simulation 1684
  - States 1436
  - Storage 1441
- Breakpoints And Markers Window 1646
- Broadcast Event
  - Add-In Model 1983
  - EA\_FileClose 1993
  - EA\_FileNew 1993
  - EA\_FileOpen 1994
  - EA\_OnPostCloseDiagram 1995
  - EA\_OnPostInitialized 1995
  - EA\_OnPostOpenDiagram 1996
  - EA\_OnPostTransform 1997
  - EA\_OnPreExitInstance 1997
  - EA\_OnRetrieveModelTemplate 1998
- BroadcastSignal Action
  - Signal Tab 866
- Browser
  - Element 687
- Build
  - Application, Analyzer menu Option 94
  - Clean Commands, Add 1416
- Build And Run Submenu
  - Package Context Menu, Project Browser 448
- Build Script
  - Create 1415
  - Deploy Script, Create 1453
  - Recursive 1416
  - Run Script, Create 1451
  - Test Script, Create 1451
- Build Scripts
  - Introduction 1415
- Build Systems Using UML
  - Enterprise Architect 7
- Built-In
  - Diagram Types 1081
  - Transformations 1307
- Business
  - Scenarios & Requirements 689
- Business Analyst
  - And Enterprise Architect 58
  - Project Role 58
- Business and Software Engineering Edition
  - Of Enterprise Architect 14
- Business Domain Model
  - Create 1204
- Business Interaction Diagram
  - Description 1191
  - Elements And Connectors 1191
  - Example 1192
- Business Model
  - Analysis 1195
- Business Modeling
  - Business Process Outline 1198
  - Events 1197
  - Example 1193
  - Goals 1198
  - Information 1196
  - Inputs 1196
  - Outputs 1197
  - Process Element 1195
  - Process Modeling Notation 1195
  - Processes 1193
  - Resources 1196
  - Traceability 493
- Business Modeling Diagram
  - Description 1191
  - Elements And Connectors 1191
  - Example 1192
- Business Process
  - Analysis 520
  - Model, Template 520
  - Outline 1198

- Business Process Execution Language (BPEL) 1235
  - Business Process Modeling 1187
  - Business Process Modeling Notation (BPMN) 1219
  - Business Rule
    - Add To Rule Task 1211
    - Create 1211
    - Element 1202
    - Export To CSV File 1211
    - Generate Code 1217
    - Modeling 1202
    - Parameters As Variables 1208
    - Remove From Rule Task 1211
    - Validate 1216
  - Business Rule Modeling
    - Business Domain Model, Create 1204
    - Business Rules 1202
    - Compose Rules 1211
    - Computation Rule Table 1211
    - Decision Table 1211
    - Export Rules To CSV File 1211
    - Generate Code 1217
    - In Class Operations 1209
    - In RuleTask Actions 1202
    - Overview 1199
    - Rule Composer 1211
    - Rule Flow Diagram 1206
    - Rule Model 1202
    - Vocabulary 1204
- C -**
- C
    - Code Generation Language Options 1531
    - Debugging, General Setup 1421
    - Import, Reverse Engineering 1510
    - Modeling Conventions 1461, 1462
    - Object Oriented Programmiing 1462
    - Options 1531
  - C#
    - Code Generation Language Options 1532
    - Import, Reverse Engineering 1510
    - Modeling Conventions 1463
    - Options 1532
    - Transformation 1307
  - C++
    - Code Generation 1533
    - Debug Symbols 1421
    - Debugging, General Setup 1421
    - Implementation Files 1533
    - Import, Reverse Engineering 1510
    - Language Options 1533
  - Modeling Conventions 1465
  - Modeling Conventions, CLI Extensions 1468
  - Modeling Conventions, Managed 1467
  - Set Up Debug Session 1420
  - Calendar
    - Allocated Resources View 396
    - Calendar Panel 390
    - Calendar Panel, Configure Event Subtypes 395
    - Diary Panel 390
    - Display Events 392
    - Options 390
    - Overview 390
    - Project 390
    - Project CalendarView 392
    - Project Tasks View 398
    - Recurring Events 392
    - Toolbar 390
  - Calibration
    - Of Project Factors 400
  - Call
    - Associate With Behaviors 716
    - Automation Interface From Enterprise Architect 1827
    - Re-associate With Behavior 716
    - Self Message 991
    - Synchronize Arguments With Behavior Parameters 716
  - Call Action
    - Pin As Argument 869
  - Call Actions 1504
  - Call Stack
    - Copy To Recording History 1450
    - Create Sequence Diagram 1450
    - Save 1450
    - View 1441
    - Window 1441
  - CallOperationAction 1504
  - Calltips
    - Code Editor, Common 1399
  - Camel Case
    - Naming Format 1341
  - Cancel
    - Default Diagram, Model 585
    - Validation 81
  - Capture State Changes
    - Setup To, Visual Execution Analyzer 1653
  - Cardinality (Multiplicity)
    - Define 776
  - CASE Tool
    - Enterprise Architect 3
  - Category
    - Add To Team Review 220

- Category
  - Create 220
  - Delete 218
- Central Buffer Node
  - Element 878
- Chaining Transformations 1306
- Change
  - BPMN Element Appearance 1228
  - Connector Source Or Target 745
  - Connector Type 744
  - Diagram Type 581
  - Element 1719
  - Element Type 86, 635
  - Elements And Requirements 1171
  - Form Of Aggregation Connector 969
  - Tracking 298
- Change Conflicts
  - Resolve 193
- Change Element
  - Hide Stereotype Letter 1721
  - Show Stereotype Letter 1721
- Change Management
  - And Requirements 1171
  - Auditing 241
  - Baselines And Differences 241
  - In Enterprise Architect 241
  - Introduction 241
  - Model Transfer 318
  - Project Data Transfer 241
  - Tracking Changes 298
  - Version Control 241, 242
- Character Set
  - Set Up For RTF Report 1773
  - Set Up For RTF Report (Legacy) 1794
- Check
  - Data Integrity 414
  - Model Integrity 414
  - Project Integrity 414
- Check Constraint
  - Create 1366
  - What Is A? 1366
- Check In
  - Explanation 278
  - Offline Packages 250
- Check Out
  - Explanation 278
  - Packages Offline 250
  - Revision Of Package 293
- Checked In Package
  - Icon 454
- Checked Out Package
  - Icon 454
- Child
  - Confirm Element As Parent 632
- Child Element
  - Paste Object As 576
- Choice
  - Element 879
- Circle Diagram Layout 605
- Circle Notation
  - For Interface Element 955
  - For Shaped Elements 938
- Class
  - Active Classes 943
  - Collection, Set 1526
  - Created In Transformation, Connect To 1338
  - Element 941
  - Elements And Connectors 554
  - Elements, Imported 1377
  - Group, Toolbox 554
  - Make Into Association Class 975
  - Members, Show/Hide On Diagram 575
  - Model Template 523
  - Parameterized Classes (Templates) 943
  - Partial 1463
  - Partial, Generate 1493
  - Reset Options 1540
  - Show Realised Interfaces On Diagram 597
  - Source Code Generation 1490
  - View 530
- Class Diagram
  - Description 798
  - Elements And Connectors 798
  - Example 799
- Class Invariant Dialog 1678
- Class Markup Selection Dialog 1646
- CLASSGUID
  - Add-In Hidden Field 1974
- Classifier
  - Behavior 904
  - Drop As Link 576
  - Drop As New Instance 576
  - Item Conveyed 984
  - Of Objects 705
  - Properties 803
  - Set 706
  - Use 706
- CLASSTYPE
  - Add-In Hidden Field 1974
- Clean
  - Data 414
  - Project 414
- Clean Commands
  - Add To Build Script 1416

- Clean Project
  - Prior To Build 1416
- Clear All Bookmarks
  - Menu Option (Edit Menu) 74
- Clear Project Clipboard
  - Menu Option (Edit Menu) 74
- Clear Selection
  - Menu Option (Edit Menu) 74
- CLI Extensions
  - C++ Modeling Conventions 1468
- Client
  - Define 782
- Client Collection
  - Automation Interface Repository 1840
- Clipboard
  - Copy Team Review Path To 218
- Clipboard File Format
  - Define 422
- Close
  - Full Screen 100
- Close Project
  - Menu Option (File Menu) 74
- CLR Versions
  - Debug .NET 1429
- Code
  - Breakpoint, Set 1438
  - Debug, Step Into Function Calls 1449
  - Debug, Step Out Of Functions 1449
  - Debug, Step Over Lines 1448
  - Debug, Step Through Function Calls 1449
  - Delete From Features In Model In Fwd Synchronization 1522
  - Generate For Business Rule 1217
  - Generated From State Machine 1498
  - Generation, Toolbar 109
  - Import, Select Language 109
  - Language, Set Default 109
  - Synchronize 1486
- Code Breakpoint
  - Set 1438
- Code Editor, Common
  - Autocompletion List 1399
  - Automatic Indentation 1397
  - Bookmarks 1396
  - Brace Matching 1397
  - Calltips 1399
  - Commenting Selections 1397
  - Configure Search Options 1404
  - Context Menu 1404
  - Create Use Case For Method Option 1406
  - Cursor History 1396
  - Customize 1394
  - Debug Tooltips 1447
  - Find And Replace 1412
  - Functions 1394
  - IME 1404
  - Input Method Editor 1404
  - Intellisense 1399
  - Key Bindings 1400
  - Line Selection 1399
  - Mouseovers 1399
  - Overview 1394
  - Scope Guides 1398
  - Search Facility 1404
  - Syntax Highlighting 1395
  - Tooltips, Debug 1447
  - Zooming 1398
- Code Engineering
  - And MDG Integration 1514
  - Broad View Of, In Enterprise Architect 1389
  - Code, Reverse Engineer 1508
  - Eclipse 1514
  - Element Context Menu 655
  - Generate Code For Single Class 1491
  - Generate Group of Classes 1492
  - Generate Package 1493
  - Generate Package Source Code 1493
  - Generate Source Code 1490
  - Introduction 1386
  - Namespaces 1495
  - Object Lifetimes 1521
  - Package Contents, Update 1494
  - Referenced XML Schema 1600
  - Reverse Engineer Source Code 1508
  - Settings 1516
  - Settings, Attribute/Operation Options 1522
  - Settings, Code Generation Constructor/Destructor Options 1521
  - Settings, Code Page for Source Editing 1523
  - Settings, General Code Options 1516
  - Settings, Import Component Types 1518
  - Settings, Source Code Options 1517
  - Synchronization 1508
  - Synchronize Model And Code 1506
  - Synchronize Package Tree 1494
  - UML Profile For XSD 1602
  - Unicode Character Set 1523
  - Update Package Contents 1494
  - Visual Studio 1514
  - With Enterprise Architect 1389
  - XML Schema 1581
  - XML Schema (XSD), Default UML To XSD Mappings 1597
  - XML Schema, Abstract XSD Models 1595

- Code Engineering
  - XML Schema, Generate XSD 1598
  - XML Schema, Import XSD 1600
  - XML Schema, Model XSD 1581
  - XSD 1581
  - XSD Datatype Packages 1610
- Code Engineering Submenu
  - Package Context Menu, Project Browser 447
- Code Generation
  - ActionScript Language Options 1530
  - Ada 2005 Language Options 1531
  - C Language Options 1531
  - C# Language Options 1532
  - C++ Language Options 1533
  - Delphi Language Options 1533
  - From Activity Diagrams 1496, 1504
  - From Behavioral Models 1496
  - From Interaction Diagrams 1496, 1503
  - From Sequence Diagrams 1496, 1503
  - From State Machine Diagrams 1496, 1497
  - Java Language Options 1535
  - Language Options 1529
  - MDG Technology Language Options 1540
  - PHP Language Options 1536
  - Python Language Options 1536
  - SystemC Language Options 1537
  - VB.NET Language Options 1537
  - Verilog Language Options 1538
  - VHDL Language Options 1538
  - Visual Basic Language Options 1539
- Code Language
  - Create C# Automatic Properties 701
  - Create Properties As Attributes 700
- Code Language Options 1520
- Code Module
  - Add To MDG Technology 1070
- Code Sections
  - Synchronize 1487
- Code Template
  - Base Templates 1483
  - Custom Templates, Create 1147
  - Default Templates 1148
  - Editor 1147, 1485
  - Editor, Add New Stereotyped Templates 1149
  - Editor, Create Templates For Custom Languages 1150
  - Editor, In SDK 1147
  - Export 1147
  - Framework, In SDK 1114
  - Framework, Overview 1482
  - Import 1147
  - Overview 1482
  - Syntax, Introduction 1115
  - Syntax, Literal Text 1115
  - Syntax, Macros 1115
  - Syntax, Template Substitution Macros 1116
- Code Template Syntax
  - Variable Definitions 1145
  - Variable References 1145
  - Variables 1145
- Codepage
  - Set Up For RTF Report 1773
  - Set Up For RTF Report (Legacy) 1794
- Collaboration
  - Element 945
  - Elements and Connectors, Now Communication 556
  - Message 998
- Collaboration Diagram
  - Description 859
  - Elements And Connectors 859
  - Example 860
  - Message Colors 861
- Collaboration Use
  - Element 946
- Collaborative Development 1523
- Collection Class
  - Additional 1532, 1533, 1535
  - Additional, Example Of Use 1528
  - Automation Interface Repository 1841
  - Example Of Use 1528
  - Set 1526
- Collections, EASL
  - Action 1138
  - Behaviors 1138
  - Classifier 1138
  - Construct 1138
  - Node 1138
  - State 1138
  - State Machine 1138
  - Transition 1138
  - Trigger 1138
  - Vertex 1138
- Color
  - Of Communication Messages 435
- Color Code External Requirements 1159
- Color Query
  - Shape Scripts 1099
- Column
  - Change Order 1352
  - Create In Data Modeling 1351
  - Data Modeling 1347
  - Definition 1351
  - In UML Data Modeling Profile 1351

- Column
  - Introduction 1347
  - Order, Change 1351
  - Properties 1351
  - Reorder 1352
  - Sequence Entries 1351
  - Stereotyped Attribute 1351
  - Unique 1351
- COM Interop
  - Debug .NET 1430
- COM Object
  - .NET Garbage Collect (Exit Method) 1855
- Combine
  - Windows In One Frame 103
- Combined Fragment
  - Create 882
  - Element 880
  - Interaction Operator 882
- Comma Separated Value
  - Export 337, 340
  - Import 337, 342
- Command
  - Deploy, Create 1453
  - Deploy, Introduction 1453
  - Run, Create 1451
  - Run, Introduction 1451
  - Unit Test, Create 1451
  - Unit Test, Introduction 1451
- Commands
  - Add To Toolbar 118
  - Change Icon Appearance 118
  - Customize 118
  - Remove From Toolbar 118
- Commenting Selections
  - Code Editor, Common 1397
- Common
  - Connectors 552
  - Elements 552
  - Group, Toolbox 552
  - Relationships 552
- Communication
  - Connector 977
  - Elements and Connectors 556
  - Group, Toolbox 556
  - Message 996
  - Message, Create 997
  - Message, Level 998
  - Message, Properties 997, 998
  - Message, Sequence 998
  - Relationship 977
- Communication Diagram
  - Description 859
- Elements And Connectors 859
  - Example 860
  - Labelled Associations 859
  - Message Colors 861
  - Numbering In 859
- Communication Message
  - Colors 435
- Communication Path
  - Connector 975
  - Relationship 975
- Community Site 30, 70
- Compact
  - Project .EAP File 418
- Compare
  - Data 337
  - DDL With Database 1380
  - Models 337
  - Projects 337
  - Utility 308, 313
- Compare Utility
  - Context Menu 316
  - Keyboard Options 316
  - Merge Options 316
  - Options 314
  - Output 315
  - Tab 315
  - Toolbar 316
- Compartment
  - Constraint 659
  - Element 659
  - Maintenance 659
  - Responsibility 659
  - Tag 659
  - Testing 659
- Compartment Events
  - Add-In Model 1986
  - EA\_GetCompartmentData 1987
  - EA\_QueryAvailableCompartments 1986
- Compiled May20 2011 3
- Complex Modeling
  - Enterprise Architect 7
- Component
  - Description 807
  - Diagram 807
  - Element 948
  - Elements And Connectors 559, 807
  - Example 808
  - Group, Toolbox 559
  - Model Template 525
  - View 530
- Compose
  - Connector 976

- Compose
  - Relationship 976
- Composite
  - Elements 647
  - Elements And Connectors 555
  - Foreign Key 1364
  - Group, Toolbox 555
  - State 917, 919
  - State Regions 820
- Composite Aggregation
  - Connector 976
  - Relationship 976
- Composite Element
  - Paste From Project Browser 578
- Composite Elements
  - Metaclass, Create With Supported Attributes 1056
- Composite Structure Diagram
  - Description 801
  - Elements And Connectors 801
  - Example 802
- Compress
  - Timeline 847
  - Transition 847
- Computation Rule Table
  - Business Rule Modeling 1211
- Concept
  - ArchMate 1272
  - BPEL 1235
  - BPMN 1219
  - Eriksson-Penker MDG Technology 1273
  - Mind Mapping 1182
  - SoaML 1624
  - SOMF 1628
  - SPEM 406
  - SysML 1552
- Concurrent Method Calls
  - In Add-Ins 1972
- Concurrent Substate
  - Regions 820
- Conditional Node
  - Structured Activity 926, 929
- Conditional Substitution
  - Field Substitution Macros, Code Template Syntax 1116
- Configuration
  - Apache Tomcat Server 1428
  - JBOSS Server 1427
  - Tomcat Server 1428
  - Tomcat Service 1428
- Configure
  - Controlled Packages With XMI 329
  - Export To XMI Stubs 329
  - Local Options 421
  - Model Validation 1689
  - Options 421
  - Package For Version Control 282
  - Packages 329
  - User Registry Settings 421
- Configure Event Subtypes
  - Project Calendar 395
- Configure Timeline Dialog
  - States Tab 840
  - Transitions Tab 842
- Confirm
  - Parent Element 632
- Connect
  - Elements 741
  - Objects 741
  - To ASA Data Repository 169
  - To Automation Interface 1823
  - To Data Repository 161
  - To Element Feature 742
  - To MSDE Server Data Repository 170
  - To MySQL Data Repository 162
  - To Oracle 10g Data Repository, Via ODBC 164
  - To Oracle 10g Data Repository, Via OLE DB 166
  - To Oracle 11g Data Repository, Via ODBC 164
  - To Oracle 11g Data Repository, Via OLE DB 166
  - To Oracle 9i Data Repository, Via ODBC 164
  - To Oracle 9i Data Repository, Via OLE DB 166
  - To PostgreSQL Data Repository 167
  - To Progress OpenEdge Data Repository 170
  - To SQL Server Data Repository 163
- Connections
  - In Relationships Window 504
  - In Team Review 230
  - To Other Team Reviews 230
  - Window 504
- Connector 756
  - Activity Edge 985
  - Add Between Locked Elements 211
  - Add Note 743
  - Add To Diagram 42
  - Add To UML Model, Quick Start 42
  - Add, Automation Interface Code Example 1956
  - Advanced Settings 761
  - Advanced, Menu Section 738
  - Aggregate 968
  - Appearance 740
  - Appearance Options On Diagram 576
  - Application 560

- Connector 756
  - Arrange 744
  - Assembly 969
  - Associate 970
  - Association 970
  - Association Class 973
  - Asynchronous Signal Message 996
  - At Page Boundaries 745
  - Automation Interface, Connector Package 1911
  - Bend At Cursor 745
  - Bend Connector 42
  - Binding 758
  - Break Connection To Feature 742
  - Change Source Or Target 745
  - Change Type 744
  - Characteristics, Edit 738
  - Collaboration Numbers, Show 576
  - Communication 977
  - Communication Path 975
  - Compose 976
  - Composite Aggregation 976
  - Connect To Element Feature 742
  - Connector 977
  - Constraints 757
  - Context Menu 736
  - Control Flow 977
  - Copy Between Instances Of Elements From Project Browser 748
  - Create Between Elements 748
  - Create From Project Browser 748
  - Create From Toolbox 546
  - Create In MDA-Style Transformation 1338
  - Create With Element Using Quick Linker 623
  - Create With Quick Linker 624
  - Create, Same Type As Previous 83
  - Crossed 745
  - Custom Properties, Set Value 738
  - Data Flow 1184
  - Delegate 978
  - Delete 750
  - Dependency 979
  - Dependency, Apply Stereotype 979
  - Deployment 980
  - Destination Role 760
  - Details 756
  - Direction Indicator 597
  - Display Options 434
  - Duplication In Transformation 1338
  - Edge 745
  - Entity Relationship Diagram 1382
  - Extend 980
  - Extension (Profile Toolbox) 560
  - Filter Display On Diagram 489, 491
  - Generalization 981
  - Generalize 981
  - Generalize (Profile Toolbox) 560
  - Hide 750
  - Hide Type On Diagram 489, 491
  - Hide/Show 740
  - Hide/Show On Diagram 576
  - Hooped 745
  - Implements 1007
  - Include 982
  - Information Flow 983
  - Inheritance 981
  - In-place Editor Options 753
  - Interrupt Flow 985
  - Invoke 553
  - Labels 597
  - Labels, Edit 753
  - Labels, Hide/Show On Diagram 576
  - Line Color 424
  - List For Requirements 1161
  - List, On Context References Tab 686
  - Locked 211
  - Manifest 986
  - Message 986
  - Move 44, 744
  - Multiplicity 758
  - Nesting 1002
  - Non-Navigable Ends, Show 576
  - Notation, Set 576
  - Notelink 1003
  - Notes 769
  - Object Flow 1003
  - Occurrence 1005
  - Off Page 745
  - Off-Page 736
  - Overview 966
  - Package Import 1006
  - Package Merge 1006
  - Page Boundary 745
  - Pkg Import 1006
  - Pkg Merge 1006
  - Precede 553
  - Properties 756
  - Properties Menu 737
  - Realization, Common Diagram 1161
  - Realization, Quick Generation Of 1161
  - Realize 1007
  - Recursion 1008
  - Redefinition 560
  - Relationship 977
  - Representation 1010

- Connector 756
  - Represents 1009
  - Reverse Direction 753
  - Role Binding 1008
  - Role, Context Menu 736
  - Select By Property 489
  - Self-Message 989
  - Shape Script Properties 1099
  - Source Role Properties 758
  - Stereotype Labels, Show On Diagram 576
  - Styles 739, 745
  - SysML Activity 1553
  - SysML Block Definition 1556
  - SysML Interaction 1558
  - SysML Internal Block 1559
  - SysML Model 1561
  - SysML Parametric 1563
  - SysML Requirement 1564
  - SysML State Machine 1565
  - SysML Use Case 1567
  - Tagged Value (Profile Toolbox) 560
  - Tagged Value, Use 1046
  - Tagged Values 760
  - Target Role Properties 760
  - Tasks 740
  - Template Binding 1010
  - To Class Created In Transformation 1338
  - Trace 1012
  - Transform 1338
  - Transition 1013
  - Type Specific Menu Section 737
  - Type, Change 744
  - Usage 1014
  - Use 1014
  - Visibility 740, 749, 751
  - What Is A? 966
  - Working With 736
- Connector Package
  - Connector, Automation Interface 1911
  - ConnectorConstraint, Automation Interface 1915
  - ConnectorEnd, Automation Interface 1916
  - ConnectorTag, Automation Interface 1918
  - RoleTag, Automation Interface 1919
- Connector Package Diagram
  - Automation Interface 1910
- Connector Styles
  - Auto Routing 745
  - Bezier 745
  - Custom 745
  - Direct 745
  - Lateral 745
  - Line Style 745
  - Set 745
  - Tree Style 745
- ConnectorConstraint
  - Automation Interface, Connector Package 1915
- ConnectorEnd
  - Automation Interface, Connector Package 1916
- ConnectorTag
  - Automation Interface, Connector Package 1918
- Console
  - Commands, Scripting Window 1820
  - Tab, Scripting Window 1820
- Constant
  - Custom, For RTF Reports 1772
- ConstLayoutStyles Enum
  - Automation Interface 1832
- Constraint
  - Attach To Connector 624
  - Automation Interface, Element Package 1876
  - Compartment, Element 659
  - Default, For SQL Server Table 1373
  - Delete 699
  - Element 666
  - In Connector 757
  - In Scenarios & Requirements Window 689
  - Inherited, Show 585
  - Internal, Import As Test 1706
  - Note, Element 913
  - Of Attributes, Create 699
  - Post Condition On Actions 871
  - Post-Condition, RTF Reporting 1741
  - Precondition On Actions 871
  - Pre-Condition, RTF Reporting 1741
  - Profile 1046
  - RTF Report Template Sections 1741
  - Status Type, Define 785
  - Stereotype 1046
  - Synchronize, And Tagged Values 1029
  - Synchronize, From MDG Toolbox Pages 1029
  - Tab, Scenario 686
  - Type, Define 784
- Contents
  - Table, MS Word, Add 1798
- Contents Submenu
  - Package Context Menu, Project Browser 448
- Context Diagram 1184
- Context Element
  - Highlight 645
  - In Multiple Selection 645
- Context Item Events
  - Add-In Model 1989
  - EA\_OnContextItemChanged 1990, 1992

- Context Item Events
  - EA\_OnContextItemDoubleClicked 1991
- Context Menu
  - Apply «Stereotype» Option 546, 1029
  - Attribute, Project Browser 452
  - Code Editor, Common 1404
  - Configure Code Editor Search Options 1404
  - Connector 736
  - Connector Label 597
  - Connector Role 736
  - Diagram 538
  - Diagram, Project Browser 451
  - Element 648
  - Element Label 597
  - Element, Add Supporting Diagrams and Elements 651
  - Element, Multiple Selection 658
  - Element, Project Browser 448
  - Item, Structural Specification 672
  - Linked Document Editor 731
  - Main 73
  - Method, Project Browser 452
  - Model Views 467
  - Model, Project Browser 443
  - New Element Or Connector 540
  - Operation, Project Browser 452
  - Package, Project Browser 444
  - Project Browser 443
  - Scripting Window 1818
  - Selected Text, Structural Specification 674
  - Structural Specification Entry Points 675
  - Team Review Options 218
- Context Reference
  - Add Comments, On Context References Tab 686
  - Add, On Context References Tab 686
  - Delete, On Context References Tab 686
  - List, On Context References Tab 686
  - Tab 686
- Continuation
  - Element 920, 921
- Control
  - Create 1290
  - Element 1289
- Control Flow
  - Connector 977
  - Guard 977
  - Relationship 977
  - Weight 977
- Control Macros
  - Code Template Syntax 1133
- Control Recording
  - Execution Analysis, Recording Sequence Diagrams 1648
- Controlled Package 282
  - Batch Export To XMI 332
  - Batch Import From XMI 333
  - Disconnect 331
  - Load 332
  - Menu, XMI 328
  - Recovery 334
  - Report Deletion Of Cross Package References In XMI 335
  - Version Control 334
  - With XMI 327
- Converge Diagram Layout 615
- Convert
  - Linked Element To Local Copy 646
  - Names In MDA Transformations 1341
- CONVERT\_DB\_TYPE 1340
- CONVERT\_NAMES
  - Macro 1341
- CONVERT\_TYPE 1340
- Convey
  - Information Item 984
- Copy
  - Attributes Between Elements 646
  - Base Project 146
  - Diagram Image To Clipboard 583
  - Diagram Image To Disk File 582
  - Diagram, Deep 583
  - Diagram, Shallow 583
  - Element Between Packages 634
  - Element On Diagram 579
  - Elements Between Diagrams 44
  - Existing Project 147
  - Maintenance Item Between Categories 1716
  - Model 140, 141, 142
  - Operations Between Elements 646
  - Package 535
  - Packages Between Projects 344
  - RTF Bookmark To Clipboard 447, 451
  - Test Between Categories 1703
  - UML Diagram, Deep 583
  - UML Diagram, Shallow 583
- Copyright Notice 19
- CORBA
  - MDG Technology For, Enterprise Architect 1036
- Co-Region Notation 996
- Corporate Edition
  - Of Enterprise Architect 14
- Correct Spell Checked Words 377
- Corrupt EAP file 419

- 
- Create 1156
    - A Project In Enterprise Architect 37
    - Access 2007 Repository 149
    - Activity, BPEL 1244
    - Adaptive Server Anywhere Repository 152
    - Add-In 1968
    - Attribute Properties 700
    - Automatic Properties, C# 701
    - Baselines 312
    - Boundary Element 1289
    - BPEL Assignments 1250
    - BPEL Model 1238
    - BPEL Web Service 1251
    - Build Script 1415
    - Check Constraint 1366
    - Columns In Data Modeling 1351
    - Combined Fragment 882
    - Communication Messages 997
    - Composite Foreign Key 1364
    - Connector With Quick Linker 624
    - Control Element 1290
    - Custom Diagram Background, Image Manager 595
    - Custom Tagged Values 1113
    - Custom View, Add-In Model 2041
    - Data Repository 148
    - Design Master, Replication 190
    - Diagram 568
    - Diagram From Linked Document 732
    - Diagram, Automatically 534
    - Document Artifact 730
    - Documents 1724
    - Element And Connector With Quick Linker 623
    - Element From Linked Document 732
    - Element From Maintenance Item 1717
    - Element From Text File 626
    - Element From Text File Text 1156
    - Element In Diagram 626
    - Element On Diagram From Project Browser 576
    - Element Template 643
    - Elements 626
    - End Event, BPEL 1247
    - Entity 1291
    - Favorites Folder, Model View 466
    - Favorites Folder, Model View (Context Menu) 467
    - Foreign Key 1361
    - Gateway, BPEL 1246
    - Hidden Submenu In Toolbox Profile 1075
    - HTML Report 1802
    - Index (Data Modeling) 1366
    - Intermediate Event, BPEL 1242
    - Learning Center Profiles 1082
    - Link Between Elements 748
    - Linked Document Template 734
    - Masked Tagged Values 1113
    - MDG Technologies, Overview 1064
    - Model 145
    - MSDE Server Repository 152
    - MySQL Repository 149
    - Notes 639
    - Oracle 10g Server Repository 151
    - Oracle 11g Server Repository 151
    - Oracle 9i Server Repository 151
    - Pattern 1022
    - Pool, BPEL 1249
    - Post In Team Review 222
    - PostgreSQL Repository 151
    - Primary Key 1358
    - Primary Key Name Template 1358
    - Profiles 1041
    - Progress OpenEdge Repository 153
    - Project 145
    - Project File 136
    - Reference Data Tagged Values 1113
    - Relationship Using Matrix 500
    - Replicas 190
    - Requirement From Text File Text 1156
    - Requirements 1156
    - Rich Text Format Report (Enhanced Generator) 1727
    - Root Node, Model View 466
    - Root Node, Model View (Context Menu) 467
    - RTF Report (Enhanced Generator) 1727
    - RTF Style Template 1734
    - RTF Style Template (Legacy) 1792
    - Search Definition 484
    - Sequence Flow, BPEL 1248
    - Slideshow, Model View 466
    - Slideshow, Model Views 471
    - Sorted Lookup Table 1366
    - SQL Server Repository 150
    - Start Event, BPEL 1240
    - Structured Tagged Values 1111
    - Table in Data Modeling 1347
    - Team Review Category 220
    - Team Review Topic 221
    - Technology Files (Option) 90
    - Text 639
    - Timing Diagram 832
    - Timing Message 1000
    - Toolbox Profile For MDG Technology 1074
    - Trigger Operation 1366

- Create 1156
  - UML Diagram 568
  - UML Pattern 1022
  - Unique Constraint 1366
  - View, Data Modeling 1370
  - Views Folder, Model View 466
  - Views Folder, Model View (Context Menu) 467
  - Views, Model View 466
  - Views, Model View (Context Menu) 467
- Create Property Implementation Dialog 700
- CreateBaselineFlag Enum
  - Automation Interface 1832
- CreateModelType Enum
  - Automation Interface 1832
- CreateObjectAction 1504
- CreateOperationAction 1504
- creating a breakpoint or recording marker on 1404
- Cross Package References
  - XMI, Report Deletion 335
- Cross Reference
  - Between EMX Files 324
  - Delete 630
  - Set For Element 630
  - Trace With Traceability Window 495
  - Use In Element 630
- Crossed
  - Connectors 745
- CSV
  - Export 337, 340
  - Export From Relationship Matrix 501
  - Export From Rule Composer 1211
  - Export State Machine Table To 828
  - Import 337, 342
  - Import Requirement Hierarchies 1162
  - Import Requirements 1162
  - Preserve Hierarchy 340
  - Specifications 338
- CTF
  - In SDK 1114
  - Overview 1482
- Current Connector
  - Toolbar 114
- Current Element
  - Toolbar 112
- Cursor History
  - Code Editor, Common 1396
- Custom
  - Diagram 1174, 1281, 1711
  - Diagram Types 1080
  - Elements and Connectors 562
  - Group, Toolbox 562
  - Stereotypes 1039
- Custom Background
  - Create For Diagram, Image Manager 595
- Custom Diagram
  - Description 1183
  - Elements And Connectors 1183
  - Model 1183
- Custom Language
  - Create Templates For In Code Template Editor 1150
  - Settings, RTF Report Generation 1773
- Custom Reference
  - Delete 630
  - Set For Element 630
  - Trace With Traceability Window 495
  - Use In Element 630
- Custom Tagged Values
  - Create 1113
- Custom Template
  - Create From Project 145
- Custom Tools
  - External Applications 121
- Custom View
  - Add-In Model 2040
- Customize
  - Commands 118
  - Diagram Appearance 569
  - Dialog 118
  - External Tools, Pass Parameters To 123
  - Keyboard 118
  - Keyboard Shortcuts 124
  - Menu Appearance 125
  - Menus 118
  - Model Search 480
  - Options 118
  - RTF Language 1773
  - RTF Language (Legacy) 1794
  - Toolbar Option Appearance 125
  - Toolbars 118, 119
  - Tools 118, 120
  - Visibility Of Elements 638
  - Window 118
- CustomProperties Collection
  - Automation Interface, ElementFeatures Package 1902
- Cut And Paste
  - Elements Between Diagrams 44
- CVS
  - Package Version Control History, In 293
  - Version Control Options 272

**- D -**

- Dashed Border
  - On Element 632
- Data
  - Breakpoint, Set 1438
  - Compare 337
  - Export 237
  - Import 239
  - Integrity 414
  - Integrity Check 414
  - Integrity, Run SQL Patches 418
  - Model Template 524
- Data Breakpoint
  - Set 1438
- Data Distribution Service
  - MDG Technology For, Enterprise Architect 1036
- Data Flow
  - Concepts 1184
  - Connector 1184
  - Context Diagram 1184
  - Diagram 1184
  - MDG Technology 1184
  - Relationship 1184
  - Toolbox Page 1184
- Data Management
  - Project Compare 93
  - Project Integrity 93
  - Project Transfer 93
  - Submenu (Tools Menu) 93
- Data Model
  - Physical 1346
  - To ERD Transformation, MDA-Style Transform 1310
  - Transformation From Entity Relationship Diagram 1317
  - Transformation To Entity Relationship Diagram 1310
- Data Modeling
  - Check Constraint 1366
  - Compare DDL With Database 1380
  - Create Columns 1351
  - Create Table 1347
  - Data Model Diagram 1374
  - Data Type Conversion Procedures 1356
  - Data Types, Introduction 1353
  - DBMS Conversion Procedure Package 1356
  - DBMS Data Types 1353
  - DDL, Generate 1379
  - Elements And Connectors 566
  - Foreign Keys 1361
  - Generate DDL 1379
  - Generate DDL For A Package 1380
  - Group, Toolbox 566
  - Index 1366
  - Introduction 1276
  - Primary Key Extended Properties 1360
  - Primary Key, Create 1358
  - Profile (UML) 1276
  - Set Database Type 1348
  - Set MySQL Table Type 1349
  - Set Oracle Table Properties 1350
  - Set Schema Owner 1349
  - Set Table Owner 1349
  - Set Table Properties 1348
  - Sorted Lookup Table 1366
  - Stored Procedure 1366, 1367
  - Tables And Columns 1347
  - Trigger Operation 1366
  - Typical Tasks 1276
  - Unique Constraint 1366
- Data Repository
  - Adaptive Server Anywhere, Connect To 169
  - Connect To 136, 161
  - Create 148
  - MSDE Server, Connect To 170
  - MySQL, Connect To 162
  - Oracle 10g, Connect To, Via ODBC 164
  - Oracle 10g, Connect To, Via OLE DB 166
  - Oracle 11g, Connect To, Via ODBC 164
  - Oracle 11g, Connect To, Via OLE DB 166
  - Oracle 9i, Connect To, Via ODBC 164
  - Oracle 9i, Connect To, Via OLE DB 166
  - PostgreSQL, Connect To 167
  - Progress OpenEdge, Connect To 170
  - SQL Server, Connect To 163
- Data Source
  - Select 1377
- Data Store
  - Element (Data Flow Diagram) 1184
- Data Transfer
  - Between Repositories 343
  - Compare Projects 337
  - Copy Packages Between Projects 344
  - Transfer Project Data 343
  - Upsize to MySQL 172
- Data Type
  - Add 777
  - Code 777
  - Conversion Procedures 1356
  - Data Modeling, Introduction 1353
  - DBMS 1353

- Data Type
  - Definition 326
  - Delete 777
  - Element 948
  - Extend 777
  - Instance 948
  - Modify 777
  - Programming Language 777
  - Referenced 948
- Database
  - Compare Package DDL With 1380
  - Default 1519
  - Design 1276
  - Keys 1276
  - Model Template 524
  - Modeling 1276
  - Schema, Import Of 1276
  - Set Default 109
  - Supported Types 1276
  - View, Report On 1370
- Database Administrator
  - And Enterprise Architect 67
  - Project Role 67
- Database Engineering
  - Introduction 1345
  - Physical Data Model 1346
  - Submenu (Tools Menu) 92
- Database Modeling
  - Enterprise Architect 7
- Database Object
  - Select From ODBC Data Source 1377
- Database Operation Properties
  - Dialog 1360
- Database Repository
  - Access Permissions For 148
  - Connect To 148
  - Create 148
  - Set Up 148
- Database Schema
  - Description 1278
  - Diagram 1278
  - Elements And Connectors 1278
  - Example 1278
- Database Table
  - Select From ODBC Data Source 1377
- Database Type
  - Set For Table 1348
- Datstore
  - Element, Activity Diagram 885
- Datatype
  - Automation Interface Repository 1843
- DBMS
  - Conversion Procedures 1356
  - Data Type Conversion Procedures 1356
- DBMS Conversion
  - Mapper 1356
  - Procedure 1356
  - Table Conversion Between DBMS Types 1356
- DDL
  - Compare With Database 1380
  - Data Modeling 1379
  - Default Script Editor 1519
  - Display In Source Code Viewer 1408
  - Generate 1379
  - Generate For Package 1380
  - Generate For Table 1379
  - Import Schema From ODBC 1375
  - Schema, Import From ODBC 1375
  - Scripts And Generated Tables 1276
  - Transformation 1310
- DDS
  - MDG Technology For, Enterprise Architect 1036
- Debug
  - .NET 1417, 1428
  - .NET CLR Versions 1429
  - .NET With COM Interop Process 1430
  - Another .NET Process 1448
  - Application, Analyzer menu Option 94
  - ASP .NET 1431
  - Break On Variable Changing Value 1445
  - C, General Setup 1421
  - C++, General Setup 1421
  - Create Sequence Diagram, Call Stack 1450
  - Deploy Script, Create New 1453
  - Deploy Script, Introduction 1453
  - Enterprise Architect 7
  - File Search, Introduction 1454
  - File Search, Use 1454
  - Inspect Process Memory 1445
  - Java 1417, 1422
  - Java Applets In Internet Browsers 1424
  - Java Web Servers 1425, 1428
  - Java, Advanced Techniques 1423
  - Java, General Setup 1422
  - Microsoft Native Applications 1421
  - On Windows 7 And Windows Vista 1418
  - Platforms 1417
  - Recording Actions 1449
  - Run Script, Create New 1451
  - Run Script, Introduction 1451
  - Save Call Stack 1450
  - Script Search 1454
  - Search Window 1454

- Debug
  - Show Loaded Modules 1446
  - Show Output 1446
  - Step Into Function Calls 1449
  - Step Out Of Functions 1449
  - Step Over Lines Of Code 1448
  - Step Through Function Calls 1449
  - Submenu (Analyzer) 95
  - Tooltips In Code Editor 1447
  - Tracepoint Statement Output 1435
  - Under Windows Vista 1417
  - Unit Test Script, Create 1451
  - Unit Test Script, Introduction 1451
  - View Call Stack 1441
  - View Local Variables 1442
  - View Local Variables, Long Values 1443
  - View Variables In Other Scopes 1444
  - Visual Basic, General Setup 1421
  - WINE Applications 1419
- Debug & Record
  - Toolbar 116
- Debug Session
  - Debug C++ 1420
  - Java, Attach To VM 1424
  - Microsoft Native Setup 1420
  - Set Up 1417
  - Set Up For .NET 1429
  - Set Up For Microsoft Native 1420
  - Start 1447
  - Stop 1447
- Debug Symbols
  - Debug C++ 1421
  - Microsoft Native 1421
- Debugger 1663
  - Actions 1416
  - Compare Variable Snapshots 1444
  - Debug Another Process 1448
  - Detach From Process 1448
  - Frameworks 1416
  - How It Works 1416
  - Introduction 1416
  - Modules Window 1446
  - On Windows 7 And Windows Vista 1418
  - Overview 1416
  - PHP, Checklist 1434
  - PHP, Scenarios 1432
  - PHP, System Requirements 1433
  - PHP, Timeouts 1434
  - Process 1416
  - Start 1447
  - Stop 1447
  - System Requirements 1417
  - Toolbar 116
  - View Elements Of Array 1443
- Debugging Actions 1441
- Decision
  - Element 886
- Decision Table
  - Business Rule Modeling 1211
  - Rule Action Section 1211
  - Rule Binding Section 1211
  - Rule Condition Section 1211
- Deep Copy
  - Of Diagram 583
- Deep History 902
  - Change From Shallow 649
- Default
  - Code Language, Set 109
  - Database 1519
  - Database, Set 109
  - DDL Script Editor 1519
  - Hours 404
  - Model Diagram, Cancel (Menu Option) 84
  - Model Diagram, Set (Menu Option) 84
  - Project Browser Behavior 455
  - Templates 1148
  - UML To XSD Mappings 1597
  - User Diagram, Cancel (Menu Option) 84
  - User Diagram, Set (Menu Option) 84
- Default Appearance
  - Background Color 641
  - Border Color 641
  - Border Thickness 641
  - Font Color 641
  - Of An Element 655
  - Of Element 641
  - Set For Profile Stereotype Objects 1049
- Default Constraint 1373
- Default Diagram
  - Model, Cancel (Menu Option) 84
  - Model, Cancel For 585
  - Model, Set (Menu Option) 84
  - Model, Set For 585
  - User, Cancel (Menu Option) 84
  - User, Set (Menu Option) 84
- Default Fonts
  - Model 429
  - Set 429
  - User 429
- Default Hours
  - Estimation 404
  - Per Adjusted Use Case Point 404
  - Project Management 404
  - Rate 404

- Default Hours
  - Settings 404
- Default Settings
  - Options Dialog 118
- Default Templates
  - Override in Code Template Editor 1148
- Default Toolbox
  - Override In Profile 1076
- Default Tools Toolbar 107
- Defect
  - Add 1719
  - Create From Test Item 1707
  - Element 1719
  - Hide Stereotype Letter 1719
  - Show Stereotype Letter 1719
- Define
  - Author 778
  - Browser Behavior 422
  - Clients 782
  - Clipboard Image File Format 422
  - Email Exchange Server 422
  - File Directory 422
  - Foreign Key Name Template 1365
  - Home Web Site 422
  - Internet Search Engine 422
  - Learning Center Contexts 1085
  - Learning Center Toolbox 1083
  - Resources 781
  - Roles 780
  - Run-Time Variable 958
  - Stereotype As Metatype 1054
  - Stereotype Constraints 1046
  - Validation Configuration For MDG Technology 1086
- Define Menu Items
  - Create Add-In 1969
- Defined Environment Types 402
- Delegate
  - Connector 978
  - Relationship 978
- Delete
  - Category In Team Review 226
  - Connectors 750
  - Connectors, Quick Start 48
  - Diagram (Single) From Project Browser 451
  - Diagrams (Multiple) From Project Browser 451
  - Diagrams From Project browser 580
  - Diagrams, Quick Start 48
  - Element Changes 1715
  - Element Defects 1715
  - Element From Diagram 637
  - Element from Model 637
  - Element From Project Browser 637
  - Element Issues 1715
  - Element Tasks 1715
  - Elements, Impact of Auditing 307
  - Elements, Quick Start 48
  - Instance Variable 959
  - Item In Team Review 226
  - Line Points 745
  - Linked Document 733
  - Linked Document Template 734
  - Locks 207
  - Menu Options 125
  - Package Attributes From Model Document 1779
  - Package In Project Browser 537
  - Packages From Model Document 1779
  - Packages, Quick Start 48
  - Post In Team Review 226
  - Project Task 357
  - Relationship 666
  - Relationship Using Matrix 500
  - Reply In Team Review 226
  - RTF Style Template 1734
  - Team Review Category 218
  - Team Review Post 218
  - Team Review Resource 218
  - Team Review Topic 218
  - Topic In Team Review 226
  - Views 532
- Delete Selected Element(s)
  - Menu Option (Edit Menu) 74
- Deletion
  - And Auditing 307
- Delphi
  - Code Generation 1533
  - Import, Reverse Engineering 1510
  - Language Options 1533
  - Limitations 1534
  - Modeling Conventions 1469
  - Properties 1534
- Demonstration
  - Of Enterprise Architect 69
- Dependency
  - Connector 979
  - Relationship 979
  - Relationship, Apply Stereotype 979
  - Report, Generate 1783
- Dependency Report
  - In Traceability 494
- Deploy
  - Add-In 1970
  - Application, Analyzer menu Option 94

- Deploy
  - MDG Technology From Add-In 1089
  - MDG Technology From File 1089
- Deploy Command
  - Create 1453
  - Introduction 1453
- Deploy Script
  - Create 1453
  - Introduction 1453
- Deployment
  - Connector 980
  - Diagram 804
  - Elements and Connectors 559
  - Group, Toolbox 559
  - Model Template 526
  - Relationship 980
  - View 530
- Deployment and Rollout
  - And Enterprise Architect 65
  - Project Role 65
- Deployment Diagram
  - Description 804
  - Elements And Connectors 804
  - Example 806
- Deployment Spec
  - Element 949
- Derived
  - Attribute 696
  - Symbol 696
- Design
  - Patterns 1021
- Design Master 188, 417
  - Create, Replication 190
- Design Systems Using UML
  - Enterprise Architect 7
- Designate Driving Triggers
  - Model State Machine For HDL 1502
- Desktop Edition
  - Of Enterprise Architect 14
  - Upsize From 148
- Desktop Tools
  - Add 120
  - Configure 120
  - Customize 120
- Destination Role 760
- DestroyObjectAction 1504
- Developer
  - And Enterprise Architect 61
  - Forward Engineering 61
  - Project Role 61
  - Reverse Engineering 61
  - Round-Trip Engineering 61
  - Visualise Package Arrangement 61
- Device
  - Element 950
- Diagram 571
  - Activity, Description 811
  - Activity, Generate From Scenarios 677
  - Add And Manage, Automation Interface Code Example 1957
  - Add Elements Via Context Menu 540
  - Add Link To Team Review Post 227
  - Add Profile Feature 1028
  - Add Profile Object 1028
  - Add To Project 568
  - Add To UML Model, Quick Start 40
  - Advanced Menu 84
  - Alias 571
  - Alternative Image For Element 593
  - Analysis 1187
  - Anti-Aliased Rendering 427
  - Anti-Aliased Text 427
  - Appearance Options 427
  - Appearance Options, Connectors 576
  - Appearance Options, Diagram Tab 571
  - Appearance Options, Element 573
  - Appearance Options, Features 574
  - Appearance Options, General 570
  - Appearance Options, Set 569
  - Appearance Options, Visible Class Members 575
  - ArchiMate 1272
  - Attribute Details, Show 574
  - Auto Route Layout 618
  - Automatic Layout 619
  - Automatic Save 430
  - Automation Interface, Diagram Package 1921
  - Background Color 424
  - Background Color Gradient 427
  - Behavior Options 430
  - Behavioral, Overview 811
  - Box Layout 609
  - BPEL 1235
  - BPMN 1219
  - Build 540
  - Business Interaction 1191
  - Business Modeling 1191
  - Business Process 520
  - Cancel Model Default 585
  - Center-Focussed Layout 605
  - Change Type 581
  - Change Type (Menu Option) 84
  - Circular Layout 605
  - Class 523, 798

- Diagram 571
  - Class Features, Visibility 426
  - Close 540
  - Collaboration 859
  - Communication 859
  - Component 525, 807
  - Composite Structure 801
  - Connector Appearance Options 576
  - Connector Notation, Show 576
  - Context 1184
  - Context Menu 538
  - Context Menu, Project Browser 451
  - Converge Layout 615
  - Copy Image 83
  - Copy, Deep 583
  - Copy, Shallow 583
  - Create 568
  - Create Automatically 534
  - Create Custom Background, Image Manager 595
  - Create From Linked Document 732
  - Create Using Image Library 595
  - Creator 570
  - Custom 1174, 1183, 1281, 1711
  - Customize Appearance 569
  - Data Flow 1184
  - Data Model, Example 1374
  - Database Schema 524, 1278
  - Define Child Type, Supported Attributes 1057
  - Delete (Multiple) From Project Browser 580
  - Delete (Single) From Project Browser 580
  - Delete Element From 637
  - Delete From Project Browser 451
  - Delete Multiple Elements From 637
  - Deployment 804
  - Details Note 571, 587
  - Diagram 1186, 1289, 1290
  - Diagram To Package 40
  - Digraph Layout 612
  - Display Options 426
  - Diverge Layout 615
  - Divide Between Pages In RTF Reprt 571
  - Drag & Drop Elements From Project Browser 576
  - Drag Existing Package Onto 536
  - Duplicate 583
  - Element Appearance Options 573
  - Element Compartments, Show/Hide 573
  - Element Icons 659
  - Element Stereotypes, Show 573
  - Element, Copy 579
  - Element, Paste 579
  - Elliptical Layout 605
  - Entity Relationship 1382
  - Eriksson-Penker 1273
  - Exclusive Edit Lock 196
  - Extended 795
  - Extended UML 1181
  - Fan Relations Layout 617
  - Feature Return Types, Show 574
  - Feature Stereotypes, Show 574
  - Features Appearance Options 574
  - Filter Display Of Connectors 489, 491
  - Filter Display Of Elements 489, 491
  - Find In Project Browser 83
  - Find Related Elements 580
  - Frame 426, 888
  - Frame (Border) 888
  - Generate From Scenarios 676
  - Grid Size, Adjust 430
  - Hyperlink 888
  - Hyperlink To 1294
  - Increase Display Size 427, 538
  - Interaction 830, 849, 859, 861
  - Interaction Overview 861
  - Layout Options 619
  - Layout Tool 604
  - Layout, Move Sections 600
  - Legend 588
  - Lock, General (Menu Option) 84
  - Lock, Require User Lock 213
  - Lock, Security Off 602
  - Lock, User/Group Lock 210
  - Logical 798
  - Maintenance 1711
  - Make All Elements Selectable 538
  - Make Model Default 585
  - Manage Display 540
  - Managing 567
  - MDG Technology 795
  - Menu 83
  - Mind Mapping 1182
  - Modeling With 538
  - Modify Z-Order, Of Elements 538
  - Move Elements 632
  - Move Sections 600
  - Move, Impact On Element 44, 886, 895, 897, 903
  - Navigation And Selection Hotkeys 582
  - Neaten Layout 614
  - Note 587
  - Notes 570
  - Notes, Show/Hide 426
  - Object 799

- Diagram 571
- Open From Shortcut 140, 142
  - Open From Shortcut (Direct Definition) 141
  - Open Package In 585
  - Overview 795
  - Package 796
  - Page Setup 426
  - Pan And Zoom 473
  - Parametric, SysML 1568
  - Paste 583
  - Per Page Layout 611
  - Place Related Elements On Current 580
  - Print From Project Browser 451
  - Print Page Footer 571
  - Print Page Header 571
  - Profile Attributes, PData and StyleEx 1081
  - Profiles 1080
  - Properties (Diagram Menu) 83
  - Properties Dialog - Connectors Tab 576
  - Properties Dialog - Diagram Tab 571
  - Properties Dialog - Elements Tab 573
  - Properties Dialog - Features Tab 574
  - Properties Dialog - General Tab 570
  - Properties Dialog, Visible Class Members, 575
  - Properties Note, Add 587
  - Properties, Set 569
  - Property Strings, Show 576
  - Qualifiers, Show 574
  - Redo Last Action 603
  - Reference 888
  - Relationship Traceability For 495
  - Relationships, Show 576
  - Rename 581
  - Requirements 521, 1174
  - Robustness 849, 859, 1186, 1187, 1288, 1289, 1290
  - Robustness, Generate From Scenarios 683
  - Rotate Image In RTF Report 571
  - RTF Document Options 571, 1729
  - RTF Report On Elements Linked From Other Packages 1729
  - Rule Model 1202
  - RuleFlow, Generate From Scenarios 678
  - Save 83, 538
  - Save As Pattern 1022
  - Save As UML Pattern (Menu Option) 84
  - Save Automatically 430
  - Save Image Of 83
  - Save Profile 1050
  - Scale View 427
  - Schema Diagram 1278
  - Scroll Through From Diagram Toolbar 111
  - Sequence 849
  - Sequence, Generate From Scenarios 681
  - Set Page Size 600
  - Show As Diagram List 538
  - Show Diagram List As 459
  - Show Realised Interfaces For Class 597
  - Show/Hide Package Contents 537
  - Slideshow, Model View 464, 471
  - SoaML 1624
  - SOMF 1628
  - SPEM 407
  - Spring Layout 613
  - Stae Machine, Generate From Scenarios 679
  - State 815
  - State Machine 815
  - Stereotype 570
  - Structural, Overview 796
  - Swimlanes 590
  - Swimlanes Matrix 592
  - Switch Between 540
  - Synchronize With Scenario Steps 676
  - SysML 1552
  - SysML, Parametric 1568
  - Tab Context Menu 543
  - Table Owner, Show 573
  - Tabs, Close 543
  - Tabs, Switch Between 545
  - Tasks, General 567
  - Timing 830
  - Toolbar 111
  - Traceability 505
  - Types 795
  - Types, Built In 1081
  - Types, Custom 1080
  - UML 795
  - Undo Last Action 603
  - Use Case 521, 813
  - User Interface 1281
  - Version 570
  - View 540
  - View Next 600
  - View Previous 600
  - Visibility Indicators 574
  - Visible Relations (Menu Option) 84
  - Water Mark 427
  - What Is A? 795
  - Working With 538
  - WSDL Binding 1617
  - WSDL Message 1613
  - WSDL Port Type 1614
  - WSDL Service 1619
  - Zoom From Diagram Toolbar 111

- Diagram 571
  - Z-Order Element 584
- Diagram Caption Bar
  - Hide 78
  - Show 78
- Diagram Filters
  - Access 491
  - Application 489
  - Clear Effect Of 491
  - Create 491
  - Delete 491
  - Disable 491
  - Enable 491
  - Introduction 489
  - Set Effect Of 491
  - Suggested Use 489
  - Window 491
- Diagram Frame 888
- Diagram Gate
  - Element 889
- Diagram Image
  - Copy To Clipboard 583
  - Copy To Disk File 582
  - Save To Disk File 582
- Diagram List
  - Context Menu Options 459
  - Description 462
  - Filter Bar 458
  - Generate RTF Report 1727
  - Options 459
  - Show As Diagram 459
  - Show Diagram As 538
  - Toolbar 459
  - Value Grouping 458
  - View Header 458
  - Work On Elements 459
- Diagram Note
  - Insert New From Toolbar 110
- Diagram Only Report
  - Generate 1784
- Diagram Package
  - Automation Interface 1920
  - Diagram, Automation Interface 1921
  - DiagramLinks, Automation Interface 1924
  - DiagramObject, Automation Interface 1925
  - Swimlane, Automation Interface 1928
  - SwimlaneDef, Automation Interface 1927
- Diagram Tabs
  - Switch Between 545
- Diagram Toolbox 546
  - Activity Group 558
  - Analysis Group 561
  - Class Group 554
  - Common Group 552
  - Communication Group 556
  - Component Group 559
  - Composite Group 555
  - Custom Group 562
  - Data Modeling Group 566
  - Deployment Group 559
  - Interaction Group 556
  - Maintenance Group 564
  - Metamodel Group 561
  - Object Group 555
  - Profile Group 560
  - Requirement Group 563
  - Shortcut Menu 551
  - State (Machine) Group 558
  - Test Domain Group 567
  - Timing Group 557
  - Use Case Group 553
  - User Interface Group 564
  - WSDL Group 565
  - XML Schema Group 566
- Diagram Type
  - Add To MDG Technology 1068
- DiagramLinks
  - Automation Interface, Diagram Package 1924
- DiagramObject
  - Automation Interface, Diagram Package 1925
- Dialog
  - All User Permissions 202
  - Application Look 125
  - Attach To Process 1448
  - Audit Settings 300
  - Class Invariant 1678
  - Class Markup Selection 1646
  - Create Property Implementation 700
  - Database Operation Properties 1360
  - Difference 302
  - Find And Replace 1412
  - General Types 782
  - Group Properties (Script) 1819
  - Link Note To Element Feature 640
  - Maintenance 787
  - Manage Uses Context List 674
  - Model Message 381
  - New Action 866
  - Operation Constraints 1679
  - Options 421
  - Package Control Options 329
  - Paste Element 576
  - People 778
  - Project Issues 358

- Dialog
  - Qualifiers 972
  - Recent Post Options 467
  - Select <Item> 690
  - Select Attribute Type 696
  - Select Property 692
  - Set Attribute 694
  - Set Feature 694
  - Set Operation 694
  - Slideshow Properties 471
  - String Viewer 1443
  - UML Types 772
  - Version Control Settings 270
- DIB Data Access Violation 1419
- Dictionary
  - User (Spell Checker) 377
- Diff Utility 308, 313
- Differencing
  - Facility 313
  - Output 315
  - With Baselines 308, 313
- Digraph Diagram Layout 612
- Direct Substitution
  - Field Substitution Macros, Code Template Syntax 1116
- Direction Indicator
  - Connector Label 597
- Directory Structure
  - Import, Reverse Engineering 1513
- Disable
  - Add-Ins 1974
  - ArchiMate 1272
  - BPEL 1235
  - BPMN 1219
  - Data Flow Diagrams 1184
  - Entity Relationship Diagram 1382
  - Eriksson-Penker MDG Technology 1273
  - Gang Of Four Pattern Technology 1545
  - GoF Pattern Technology 1545
  - ICONIX 1542
  - MDG Technologies 1033
  - Mind Mapping 1182
  - Security 196
  - SoaML 1624
  - SOMF 1628
  - SPEM 406
  - SysML 1552
- Disable Recording Markers 1647
- Disconnect
  - Controlled Package 331
- Discussion Forum
  - Now Team Review 216
- Display
  - Attributes, Inherited 585
  - Connector Properties, Shape Scripts 1099
  - Constraints, Inherited 585
  - Element Properties, Shape Scripts 1099
  - Full Screen 100
  - Inherited Attributes 702
  - Inherited Operation 713
  - Operations, Inherited 585
  - Requirements, Inherited 585
  - Tagged Values, Inherited 585
- Display Options
  - Connector 434
  - Diagram 426
  - Element 432
  - Link 434
  - Relationship 434
- Display Size
  - Diagram, Increase 427
- Distributed Development
  - Replication 188
  - XMI Import/Export 188
- Diverge Diagram Layout 615
- Dock
  - Two Elements 655
- Dock Windows
  - Navigation Compass 103
- Dockable
  - Element Option 655
  - Windows 102
- Docked Windows
  - For Requirements 1169
- Document
  - Enterprise Architect Content 1724
  - Exclude Packages 1724
  - HTML 1724
  - Linking 729
  - Projects 1724
  - Resource, RTF Generator (Enhanced) 1766
  - Rich Text Format 1724
  - RTF 1724
  - Single Element, RTF (Legacy) 1788
  - WSDL Element 1620
- Document Artifact
  - Element 729, 951
  - Link Into RTF Report 729, 1737
- Document Generation
  - Enterprise Architect 7
- Document Generator Interface
  - Automation Interface 1947
- Document Options

- Document Options
    - RTF Generator, From Diagram (Enhanced) 1729, 1730
    - RTF Generator, From Diagram (Legacy) 1729, 1730
    - RTF Report 1767
  - Documentation
    - Elements 1774
    - Generate, Project Browser Option 452
    - Group, Toolbox 1774
    - HTML 1802
    - Rich Text Format 1726
    - RTF 1726
  - Documentation (Reports)
    - Submenu (Project Menu) 80
  - Documentation Submenu
    - Package Context Menu, Project Browser 447
    - Reports 447
  - DocumentationGeneratorInterface Package
    - DocumetationGenerator Class, Automation Interface 1947
  - DocumentBreak Enum
    - Automation Interface 1833
  - DocumentGenerator Class
    - Automation Interface 1947
  - DocumentType Enum
    - Automation Interface 1833
  - DoDAF-MODAF
    - MDG Technology For, Enterprise Architect 1036
  - Domain
    - Model Template 522
    - Organizational Relationships 522
    - Physical Units 522
    - Structure 522
  - DOORS, Telelogic
    - MDG Link For, Enterprise Architect 1036
  - Dot
    - On Association 758
  - Download
    - Gang Of Four Patterns 1023
    - GoF Patterns 1023
  - Drag
    - Element From Project Browser Onto Diagram 576
    - Objects From Project Browser Onto Diagram 576
    - Text Into Diagram To Create Element 1156
  - Drawing Methods
    - Shape Scripts 1094
  - Drop
    - Classifiers As Links 576
    - Classifiers As New Instances 576
    - Elements From Project Browser Onto Diagram 576
    - Objects From Project Browser Onto Diagram 576
  - DTD
    - Validate XMI Import/Export 326
  - Duplicate
    - Diagram 583
    - Element 634
    - Package 535
    - UML Diagram 583
  - Duration
    - Constraint 993
    - Constraint Between Messages 993
    - Observation 993
  - Dynamic View 530
  - Dynamic Visual Filters 489
- E -**
- EA\_AddinLicenseGetDescription
    - Add-In License Management Event, Add-In Model 1986
  - EA\_AddinLicenseValidate
    - Add-In License Management Event, Add-In Model 1985
  - EA\_Connect
    - Add-In Event 1977
  - EA\_Disconnect
    - Add-In Event 1977
  - EA\_FileClose
    - Broadcast Events, Add-In Model 1993
  - EA\_FileNew
    - Broadcast Events, Add-In Model 1993
  - EA\_FileOpen
    - Broadcast Events, Add-In Model 1994
  - EA\_GetCompartmentData
    - Compartment Events, Add-In Model 1987
  - EA\_GetMenuItems
    - Add-In Event 1978
  - EA\_GetMenuState
    - Add-In Event 1979
  - EA\_MenuClick
    - Add-In Event 1980
  - EA\_OnAttributeTagEdit
    - Tagged Value Broadcast Events, Add-In Model 2031
  - EA\_OnConnectorTagEdit
    - Tagged Value Broadcast Events, Add-In Model 2032
  - EA\_OnContextItemChanged

|                                             |      |  |  |
|---------------------------------------------|------|--|--|
| EA_OnContextItemChanged                     |      |  |  |
| Context Item Events, Add-In Model           | 1990 |  |  |
| EA_OnContextItemDoubleClicked               |      |  |  |
| Context Item Events, Add-In Model           | 1991 |  |  |
| EA_OnDeleteTechnology                       |      |  |  |
| Technology Events, Add-In Model             | 2038 |  |  |
| EA_OnElementTagEdit                         |      |  |  |
| Tagged Value Broadcast Events, Add-In Model | 2033 |  |  |
| EA_OnEndValidation                          |      |  |  |
| Model Validation Broadcasts, Add-In Model   | 2001 |  |  |
| EA_OnImportTechnology                       |      |  |  |
| Technology Events, Add-In Model             | 2039 |  |  |
| EA_OnInitializeTechnologies                 |      |  |  |
| Technology Events, Add-In Model             | 2035 |  |  |
| EA_OnInitializeUserRules                    |      |  |  |
| Model Validation Broadcasts, Add-In Model   | 2000 |  |  |
| EA_OnNotifyContextItemModified              |      |  |  |
| Context Item Events, Add-In Model           | 1992 |  |  |
| EA_OnOperationTagEdit                       |      |  |  |
| Tagged Value Broadcast Events, Add-In Model | 2034 |  |  |
| EA_OnOutputItemClicked                      |      |  |  |
| Add-In Event                                | 1981 |  |  |
| EA_OnOutputItemDoubleClicked                |      |  |  |
| Add-In Event                                | 1982 |  |  |
| EA_OnPostActivateTechnology                 |      |  |  |
| Technology Events, Add-In Model             | 2037 |  |  |
| EA_OnPostCloseDiagram                       |      |  |  |
| Broadcast Events, Add-In Model              | 1995 |  |  |
| EA_OnPostInitialized                        |      |  |  |
| Broadcast Events, Add-In Model              | 1995 |  |  |
| EA_OnPostNewAttribute                       |      |  |  |
| Post-New Events, Add-In Model               | 2014 |  |  |
| EA_OnPostNewConnector                       |      |  |  |
| Post-New Events, Add-In Model               | 2011 |  |  |
| EA_OnPostNewDiagram                         |      |  |  |
| Post-New Events, Add-In Model               | 2012 |  |  |
| EA_OnPostNewDiagramObject                   |      |  |  |
| Post-New Events, Add-In Model               | 2013 |  |  |
| EA_OnPostNewElement                         |      |  |  |
| Post-New Events, Add-In Model               | 2011 |  |  |
| EA_OnPostNewGlossaryTerm                    |      |  |  |
| Post-New Events, Add-In Model               | 2016 |  |  |
| EA_OnPostNewMethod                          |      |  |  |
| Post-New Events, Add-In Model               | 2015 |  |  |
| EA_OnPostNewPackage                         |      |  |  |
| Post-New Events, Add-In Model               | 2015 |  |  |
| EA_OnPostOpenDiagram                        |      |  |  |
| Broadcast Events, Add-In Model              | 1996 |  |  |
| EA_OnPostTransform                          |      |  |  |
| Broadcast Events, Add-In Model              | 1997 |  |  |
| EA_OnPreActivateTechnology                  |      |  |  |
| Technology Events, Add-In Model             | 2036 |  |  |
| EA_OnPreDeleteAttribute                     |      |  |  |
| Pre-Deletion Events, Add-In Model           | 2018 |  |  |
| EA_OnPreDeleteConnector                     |      |  |  |
| Pre-Deletion Events, Add-In Model           | 2020 |  |  |
| EA_OnPreDeleteDiagram                       |      |  |  |
| Pre-Deletion Events, Add-In Model           | 2020 |  |  |
| EA_OnPreDeleteDiagramObject                 |      |  |  |
| Pre-Deletion Events, Add-In Model           | 2021 |  |  |
| EA_OnPreDeleteElement                       |      |  |  |
| Pre-Deletion Events, Add-In Model           | 2017 |  |  |
| EA_OnPreDeleteGlossaryTerm                  |      |  |  |
| Pre-Deletion Events, Add-In Model           | 2023 |  |  |
| EA_OnPreDeleteMethod                        |      |  |  |
| Pre-Deletion Events, Add-In Model           | 2019 |  |  |
| EA_OnPreDeletePackage                       |      |  |  |
| Pre-Deletion Events, Add-In Model           | 2022 |  |  |
| EA_OnPreDeleteTechnology                    |      |  |  |
| Technology Events, Add-In Model             | 2037 |  |  |
| EA_OnPreExitInstance                        |      |  |  |
| Broadcast Events, Add-In Model              | 1997 |  |  |
| EA_OnPreNewAttribute                        |      |  |  |
| Pre-New Events, Add-In Model                | 2027 |  |  |
| EA_OnPreNewConnector                        |      |  |  |
| Pre-New Events, Add-In Model                | 2025 |  |  |
| EA_OnPreNewDiagram                          |      |  |  |
| Pre-New Events, Add-In Model                | 2026 |  |  |
| EA_OnPreNewDiagramObject                    |      |  |  |
| Pre-New Events, Add-In Model                | 2026 |  |  |
| EA_OnPreNewElement                          |      |  |  |
| Pre-New Events, Add-In Model                | 2024 |  |  |
| EA_OnPreNewGlossaryTerm                     |      |  |  |
| Pre-New Events, Add-In Model                | 2030 |  |  |
| EA_OnPreNewMethod                           |      |  |  |
| Pre-New Events, Add-In Model                | 2028 |  |  |
| EA_OnPreNewPackage                          |      |  |  |
| Pre-New Events, Add-In Model                | 2029 |  |  |
| EA_OnRetrieveModelTemplate                  |      |  |  |
| Broadcast Events, Add-In Model              | 1998 |  |  |
| EA_OnRunAttributeRule                       |      |  |  |
| Model Validation Broadcasts, Add-In Model   | 2004 |  |  |
| EA_OnRunConnectorRule                       |      |  |  |
| Model Validation Broadcasts, Add-In Model   | 2003 |  |  |
| EA_OnRunDiagramRule                         |      |  |  |
| Model Validation Broadcasts, Add-In Model   | 2002 |  |  |
| EA_OnRunElementRule                         |      |  |  |

- EA\_OnRunElementRule
  - Model Validation Broadcasts, Add-In Model 2001
- EA\_OnRunMethodRule
  - Model Validation Broadcasts, Add-In Model 2005
- EA\_OnRunPackageRule
  - Model Validation Broadcasts, Add-In Model 2002
- EA\_OnRunParameterRule
  - Model Validation Broadcasts, Add-In Model 2005
- EA\_OnStartValidation
  - Model Validation Broadcasts, Add-In Model 2000
- EA\_QueryAvailableCompartments
  - Compartment Events, Add-In Model 1986
- EA\_ShowHelp
  - Add-In Event 1983
- EAB File
  - Export 290
  - Import 291
  - Model Branch File 290
- EABase
  - As Source 146
  - Project 35
  - Project File 145
- EAEditionTypes Enum
  - Automation Interface 1833
- EAExample File 138
- EAP File
  - As Project Database 136
  - Corrupt 419
  - Iterate Through, Automation Interface Code Example 1953
- EASL
  - Behavioral Model Templates 1137
  - Code Generation Macros, Behavioral Model 1137
  - Enterprise Architect Simulation Library 1137
- EASL Collections
  - Action 1138
  - Behavior 1138
  - Classifier 1138
  - Construct 1138
  - Node 1138
  - State 1138
  - State Machine 1138
  - Transition 1138
  - Trigger 1138
  - Vertex 1138
- EASL Properties
  - Action 1141
  - Argument 1141
  - Behavior 1141
  - Call Event 1141
  - ChangeEvent 1141
  - Classifier 1141
  - Condition 1141
  - Construct 1141
  - Edge 1141
  - EventObject 1141
  - Instance 1141
  - Parameter 1141
  - Primitive 1141
  - PropertyObject 1141
  - SignalEvent 1141
  - State 1141
  - StateMachine 1141
  - TimeEvent 1141
  - Transition 1141
  - Trigger 1141
  - Vertex 1141
- EASL\_GET
  - Code Generation Macro, Behavioral Model 1137
- EASLList
  - Code Generation Macro, Behavioral Model 1137
- ECF
  - Value 402
  - Weighting 402
- Eclipse
  - MDG Integration For, Enterprise Architect 1036
  - MDG Link For, Enterprise Architect 1036
- Edit
  - Attribute Name, In-Place Editor 722
  - Element Name, In-Place Editor 722
  - Item In Team Review 226
  - Linked Document Template 734
  - Linked Documents 731
  - Menu 74
  - Operation Name, In-Place Editor 722
  - Pattern Default 1024
  - RTF Style Template 1734
  - Test Details 1696
- Editions
  - Business and Software Engineering 14
  - Corporate 14
  - Desktop 14
  - Floating Licence 14
  - Lite 18
  - Of Enterprise Architect 14
  - Of Enterprise Architect, Introduction 13
  - Professional 14
  - Standalone 14
  - System Engineering 14

- Editions
  - Ultimate 14
- Editor
  - Team Review 226
- Editor Language Properties
  - Code Editor Properties 1520
  - Global Options 1520
  - Language-Specific Options 1520
  - Macro Key Assignment 1520
- Effort
  - Attributes 1876
  - Automation Interface, Element Package 1876
  - Methods 1876
- Effort Management 350
- Effort Types 353
  - Define 353
  - Global 353
  - Non-Global 350
- EJB
  - Entity Bean Transformations 1314
  - Session Bean Transformations 1314
- Element 666, 916, 920
  - Abort Edit Changes 87
  - Accept Edit Changes 87
  - Action 865
  - Activity 873
  - Activity Final 895
  - Activity Partition 914
  - Activity Region 916
  - Actor 877
  - Add And Manage, Automation Interface Code Example 1955
  - Add Attribute, In-place Editor 726
  - Add Directly To Package 627
  - Add Link To Team Review Post 227
  - Add New Item (Inline Features Menu Option) 87
  - Add Operation, In-place Editor 726
  - Add Supporting Diagrams and Elements 651
  - Add Tagged Value (Option) 86
  - Add To Diagram 41
  - Add To Diagram From Project Browser 576
  - Add To Diagram Via Context Menu 540
  - Add To Favorites 653
  - Add To Profile 1042
  - Add To UML Model, Quick Start 41
  - Advanced Settings 664
  - Align 88, 658
  - Align From Diagram Toolbar 111
  - Align Multiple 635
  - Alternative Image 593
  - Appearance, Context Menu Option 655
  - Appearance, Format From Toolbar 541
  - Apply Image From Clipboard 88
  - ArchiMate 1272
  - Artifact 940, 941
  - Associated Files 687
  - Attributes Option 654
  - Author 662
  - Auto Counters 628
  - Auto Naming 628
  - Auto Numbering 628
  - Automatic Properties, C# 701
  - Automation Interface, Element Package 1877
  - Autosize Group 589
  - Autosize Single 589
  - Background Color 424, 425
  - Behavioral Diagram 864
  - Boundary 1288
  - Boundary, Settings 932
  - BPMN, Change Appearance 1228
  - Browser Window 687
  - Bulk Update 411
  - Business Rule 1202
  - Cardinality (Multiplicity), Non-Displayable 663
  - Central Buffer Node 878
  - Change 1719, 1721
  - Change Appearance, BPMN 1228
  - Change Type 86, 635
  - Changes And Issues 1713
  - Changes, Add/Modify/Delete 1715
  - Child Validation 1691
  - Choice 879
  - Class 941
  - Code Engineering Menu 655
  - Collaboration 945
  - Collaboration Use 946
  - Combined Fragment 880
  - Compartments 659
  - Complexity 662
  - Component 948
  - Composite 647
  - Concurrency 663
  - Confirm As Parent 632
  - Connect 741
  - Connectors In Relationships Window 504
  - Constraint Note 913
  - Constraint, Attach 87
  - Constraints 666
  - Context Menu 648
  - Context Menu, Add Submenu 651
  - Context Menu, Advanced 649
  - Context Menu, Features 654
  - Context Menu, Find Submenu 653
  - Context Menu, Project Browser 448

- Element 666, 916, 920
  - Continuation 920
  - Continutaion 921
  - Control 1289, 1290
  - Coordinates On Status Bar 117
  - Copy And Paste Between Diagrams 44
  - Copy Between Packages 634
  - Create Child Diagram 568
  - Create From Linked Document 732
  - Create From Maintenance Item 1717
  - Create From Text File 626
  - Create From Text File Text 1156
  - Create From Toolbox 546
  - Create In Diagram 626
  - Create Link From Project Browser 748
  - Create With Quick Linker 623
  - Create, Same Type As Previous 83
  - Cross References (Option) 86
  - Customize Visibility 638
  - Cut And Paste Between Diagrams 44
  - Dashed Border On 632
  - Data Store, Data Flow Diagram 1184
  - Data Type 948
  - Datastore, Activity Diagram 885
  - Decision 886
  - Default Appearance 88
  - Default Element Template 643
  - Defect 1719
  - Defects 1713
  - Defects, Add/Modify/Delete 1715
  - Delete 637
  - Delete Item From 87
  - Deployment Spec 949
  - Details 663
  - Device 950
  - Diagram Frame 888
  - Diagram Gate 889
  - Display Depth 584
  - Display Options 432
  - Dockable 655
  - Document Artifact 951
  - Documentation 1774
  - Drag From Project Browser 576
  - Duplicate 634
  - Edit Attribute Keyword 724
  - Edit Attribute Scope 723
  - Edit Attribute Stereotype, In-place Editor 722
  - Edit Item, Tasks 720
  - Edit Name, In-Place Editor 722
  - Edit Operation Parameter Keyword 724
  - Edit Operation Parameter Kind 725
  - Edit Operation Scope 723
  - Edit Operation Stereotype, In-place Editor 722
  - Embedded 86
  - Embedded Submenu 653
  - Embedded, Add 654
  - Endpoint 890
  - Entity 1290
  - Entity, Entity Relationship Diagram 1382
  - Entry Point 891
  - Enumeration 952
  - Eriksson-Penker 1273
  - Event 1292
  - Exception 892
  - Execution Environment 952
  - Exit Point 895
  - Expansion Region 893
  - Export Data In CSV Format 340
  - Expose Interface 953
  - Extended By Stereotype 1288
  - External Requirements 665
  - External, Data Flow Diagram 1184
  - Feature 1166
  - Feature, Connect To 742
  - Feature, Disconnect From 742
  - Fill Color 424, 425
  - Fill Color Gradient 427
  - Filter Display On Diagram 489, 491
  - Find In Diagram Menu Options 448
  - Find in Diagrams 85, 653
  - Find In Diagrams, Package Browser 459
  - Find In Project Browser 85, 653
  - Find Related 580
  - Fine Movement 600
  - Flow Final 897
  - Font, Set 88
  - Fork 898, 900
  - Fragment 880
  - Gang Of Four Pattern 1545
  - Gap, Model 510
  - General Settings 662
  - Get Project Custom Colors 642
  - GoF Pattern 1545
  - Hide Type On Diagram 489, 491
  - Highlight Context 645
  - History 902
  - Hyperlink 1292, 1294
  - Icon, Project Browser, User-Defined 1051
  - Icons On Diagram 659
  - Image 1295
  - Import Data In CSV Format 342
  - Include Linked In RTF Report 1729, 1730
  - Information Item 954
  - Initial 903

- Element 666, 916, 920
  - In-place Editor Options 720
  - In-Place Formatting 659
  - Insert Maintenance Feature 727
  - Insert New Feature (Inline Features Menu Option) 87
  - Insert Operation Parameter 726
  - Insert Related Elements 652
  - Insert Testing Features 728
  - Instance 957
  - Interaction 904
  - Interaction Occurrence 906
  - InteractionUse 906
  - Interface 955
  - Internal Requirements 664
  - Interruptible Activity Region 907
  - Issue 1719
  - Issues And Changes 1713
  - Issues, Add/Modify/Delete 1715
  - Join 898, 901
  - Junction 908
  - Keywords 662
  - Labels 597
  - Layout 658
  - Legend 588
  - Lifeline 910
  - Line Color 425
  - Link To Attribute Via Object 704
  - Linked, Convert To Local Copy 646
  - Lock Indicators 214
  - Lock, Require User Lock 213
  - Lock, Security Off 655
  - Lock, User/Group Lock 210
  - Locked, Add Connector To 211
  - Maintenance 1713
  - Make All Selectable/Unselectable 538
  - Make Non-Selectable, Multiple Elements 658
  - Make Non-Selectable, Single Element 655
  - Make Same Size 88
  - Make Selectable, Multiple Elements 658
  - Make Selectable, Single Element 655
  - Managing 625
  - Master Document 1774
  - Match Size 658
  - Menu 85
  - Merge 886
  - Merge Node 910
  - Mind Mapping 1182
  - Model Document 1774
  - Modeling With 625
  - Move Between Packages 633
  - Move By Increments 600
  - Move In Diagrams 632
  - Move, Impact Of Diagram 44
  - Multiple Selection 658
  - Multiple Update 411
  - N-Ary Association 1296
  - N-Ary, Entity Relationship Diagram 1382
  - New 626
  - Node 956
  - Non-Selectable, Multiple Elements 658
  - Non-Selectable, Single Element 655
  - Note (Constraint, Comment) 913
  - Note, Attach 87
  - Notes 639, 769
  - Nudge 88, 600
  - Object 957
  - Occurrence 906
  - Operations Option 654
  - Overrides & Implementations 86
  - Package 960
  - Packaging Component 1297
  - Parents And Interfaces 86
  - Part 960
  - Partition 914
  - Paste As Link 75
  - Paste As New 75
  - Paste Copy 634
  - Paste From Clipboard As Metafile 75
  - Paste From Project Browser 576
  - Phase 662
  - Phase, Update For Package 411
  - Place Related On Current Diagram 580
  - Port 962
  - Position Options 85
  - Postconditions 666
  - Preconditions 666
  - Primitive 964
  - Process 1195, 1297
  - Process, Data Flow Diagram 1184
  - Properties Dialog 660
  - Properties Dialog, Changes and Defects 1722
  - Properties Dialog, General Settings 662
  - Properties Menu 649
  - Properties Window 689
  - Properties, As Attributes 700
  - Properties, Edit From Package 796
  - Properties, Links 666
  - Pseudo-State 820
  - Receive 915
  - Receive Event 1292
  - Region 916
  - Region, Expansion 893
  - Region, Interruptible Activity 907

- Element 666, 916, 920
  - Relationship Traceability For 495
  - Relationship, Delete 666
  - Relationship, Entity Relationship Diagram 1382
  - Relationship, Hide 666
  - Relationship, Show 666
  - Requirement 1154, 1164
  - Requirements 664
  - Resize 636
  - Responsibilities 664
  - Risk 1298
  - Rule Task 1206
  - Scenario 667
  - Scenarios & Requirements 689
  - Screen 1281
  - Select By Property 489
  - Selectable, Multiple Elements 658
  - Selectable, Single Element 655
  - Send 917
  - Send Event 1292
  - Sequence Diagram 853
  - Sequence, Lifecycle 851
  - Set Alternative Image 88
  - Set Cross References 630
  - Set Custom References 630
  - Set Default Appearance 641
  - Set Element Template Package 643
  - Set Feature Visibility 654
  - Set Font 657
  - Set Parent 628
  - Set Project Custom Colors 642
  - Shadow Color 424
  - Shape Script Properties 1099
  - Show Usage 630
  - Signal 964
  - Size 636
  - SoaML 1624
  - SOMF 1628
  - SPEM 407
  - State 917
  - State Invariant 920, 922
  - State Lifeline 923
  - State Machine 925
  - State/Continuation 920
  - Status 662
  - Status, Update For Package 411
  - Stereotype 662
  - Stored Procedure 1367
  - Structural Diagram 940
  - Structured Activity, Conditional Node 926, 929
  - Structured Activity, Loop Node 926, 928
  - Structured Activity, Sequential Node 926, 927
  - Structured Activity, Structured Node 926, 927
  - Sub-Activity 873
  - Sub-Activity, Conditional Node 926, 929
  - Sub-Activity, Loop Node 926, 928
  - Sub-Activity, Sequential Node 926, 927
  - Sub-Activity, Structured Node 926, 927
  - Submachine State 917, 925
  - Superimposition 584
  - Synch 930
  - Synchronize Stereotyped Tagged Values From Toolbox 546
  - Synchronize Stereotypes With Profile 1029
  - SysML Activity 1553
  - SysML Block Definition 1556
  - SysML Interaction 1558
  - SysML Internal Block 1559
  - SysML Model 1561
  - SysML Parametric 1563
  - SysML Requirement 1564
  - SysML State Machine 1565
  - SysML Use Case 1567
  - System Boundary 931
  - Table 1277
  - Task 1299
  - Tasks 625, 1713
  - Tasks, Add/Modify/Delete 1715
  - Template Package 643
  - Template Parameters 663
  - Template, Default Element 643
  - Templates And Profiles 643, 1026
  - Terminate 933
  - Test Case 1674
  - Test Cut 1675
  - Test Scripts Compartment 1707
  - Test Set 1675
  - Test Suite 1676
  - Text 639
  - Text Color 424, 425
  - Toolbar 110
  - Transformation 1305
  - Trigger 934
  - Type, Change 86
  - UI Control 1282
  - UML 864
  - Usage 630
  - Use Case 935
  - Use Circle Notation 938
  - Use Cross References 630
  - Use Custom References 630
  - Use Extras, Automation Interface Code Example 1958
  - Use Rectangle Notation 938

- Element 666, 916, 920
  - User Interface 1282
  - Value Lifeline 938
  - Version 662
  - Version, Update For Package 411
  - View Properties (Inline Features Menu Option) 87
  - View, Data Modeling 1370
  - Visibility 663
  - Visibility Options 434
  - Visual Representation, Introduction 658
  - Work On From Diagram List 462
  - Work On From Package Browser 456
  - Work On From Toolbar 112
  - Working With 625
  - WSDL Binding 1617
  - WSDL Message 1613
  - WSDL Service 1619
  - WSDL, Document 1620
  - WSDL, Port Type 1614
  - Z-Order 88, 584
- Element Context Menu
  - Add Submenu, Project Browser 450
- Element Feature
  - Attribute 695
  - Connect To 742
  - Disconnect From 742
  - Operation 707
- Element Package, Automation Interface
  - Constraint 1876
  - Diagram 1873
  - Effort 1876
  - Element 1877
  - File 1887
  - Issue 1888
  - Metric 1889
  - Requirement 1890
  - Resource 1891
  - Risk 1891
  - Scenario 1892
  - ScenarioExtension 1893
  - ScenarioStep 1894
  - TaggedValue 1895
  - Test 1896
- Element Templates
  - And Profiles 1039
- Element View
  - Project Task Allocation 368
- ElementFeatures Package, Automation Interface 1909
  - Attribute Class 1898
  - AttributeConstraint 1900
  - AttributeTag 1901
  - CustomProperties Collection 1902
  - Diagram 1897
  - EmbeddedElements Collection 1902
  - Method 1903
  - MethodConstraint 1905
  - MethodTag 1906
  - Parameter 1906
  - ParamTag 1908
  - Partitions Collection 1908
  - Properties 1909
  - Property 1909
  - Transitions Collection 1910
- Ellipse Diagram Layout 605
- Email
  - Access Within Enterprise Architect 128
  - Exchange Server, Define Default 422
- Embedded Elements
  - Dialog 654
  - Element Context Submenu 653
  - Include When Pasting Composite 578
  - Incorporate Inherited Properties 654
  - Maintain Layout In Pasting Composite 578
  - Submenu Option 86
  - Window 654
- EmbeddedElements Collection
  - Automation Interface, ElementFeatures Package 1902
- EMX
  - File Cross References 324
  - File Import 324
- Enable
  - Add-Ins 1974
  - ArchiMate 1272
  - Data Flow Diagrams 1184
  - Entity Relationship Diagram 1382
  - Eriksson-Penker MDG Technology 1273
  - Exclusive Diagram Edit Lock 196
  - Gang Of Four Pattern Technology 1545
  - GoF Pattern Technology 1545
  - ICONIX 1542
  - MDG Technologies 1033
  - Mind Mapping 1182
  - Security 196
  - SoaML 1624
  - SOMF 1628
  - SPEM 406
- Encrypt Password
  - Prior To Release 7.1 Of Enterprise Architect 208
- Encrypt Password (Repository)

- Encrypt Password (Repository)
  - At Release 7.1 Of Enterprise Architect 140, 143
- End Event, BPEL
  - Create 1247
  - Model 1247
  - Types 1247
- End User License Agreement 20
- Endpoint
  - Element 890
- Enterprise Architect
  - Add-In Model 1967
  - Alignment With UML 793
  - And Deployment 65
  - And IDEs 7
  - Autohide Windows 105
  - Automation 1816
  - Build Systems 7
  - CASE Tool 3
  - Change Control, Project 1710
  - Change Management 241
  - Code Engineering With 1389
  - Community Site 30, 70
  - Connectors 966
  - Create Project, Tutorial 37
  - Database Modeling 7
  - Debug 7
  - Demonstration 69
  - Design Systems 7
  - Dock Windows 103
  - Dockable Windows 102
  - Editions, Differences Between 14
  - Editions, Introduction 13
  - Editor 1519
  - End User License Agreement 20
  - Example Project File 138
  - Export Data In CSV Format 340
  - Extend Trial Period 13
  - Feedback Pages 3
  - For Business Analysts 58
  - For Database Administrators 67
  - For Developers 61
  - For Implementation Managers 65
  - For Project Managers 63
  - For Software Architects 59
  - For Software Engineers 60
  - For Technology Developers 65
  - For Testers 64
  - Formal Statements 19
  - Fundamental Processes 5
  - Generate Documentation 7
  - Generate Source Code 7
  - Getting Started 34
  - Glossary 2057
  - Help 30
  - How To Use 3
  - Import Data In CSV Format 342
  - In Action 69
  - Install 27
  - Interfaces For Modeling 69
  - Introduction 3
  - Key Features 10
  - Keyboard Shortcuts 128
  - Keyboard/Mouse Shortcuts 133
  - License Agreement 20
  - Lite Edition 18
  - Main Context Menu 73
  - Main Menu 73
  - Manage Model Structure 7
  - Manage Requirements 7
  - MDA Transformation 7
  - Model Complexity 7
  - Model Requirements 7
  - Modeling Tool 3
  - Navigation Compass 103
  - Object Model, Introduction 1822
  - Online Resources 70
  - Online User Guide 3
  - Order 26
  - Performance, DBMS Connections 162, 163, 164, 166, 167, 169, 170
  - Performance, In Auditing 298
  - Performance, WAN Optimizer 183
  - Predefined Search Definitions 479
  - Pricing And Purchasing 26
  - Professional Roles 56
  - Project Change Control 1710
  - Project Files, Open A Project 138
  - Project Maintenance 1710
  - Project Roles 56
  - Project Team Review 216
  - Project, What Is A? 137
  - Quality Control 1687
  - Quickstart Tutorial 37
  - Read-Only Edition 18
  - Register Full License 29
  - Replication Merge Rules 188
  - Reverse Engineer 7
  - Scripting 1816
  - SDK, Introduction 1038
  - Select Edition To Trial 13
  - Share Model Development 7
  - Software Product License Agreement 20
  - Spell Checking 372

- Enterprise Architect
  - Start 35
  - Start Page 70
  - Support 30, 31
  - Testing 1687
  - Tools And Features For Modeling 69
  - Trial Version 13
  - UML 2.3 Support 7
  - UML Modeling With 515
  - User Interface 69
  - Uses Of 7
  - Visualize Systems 7
  - Web Services Access 128
  - What Can I Do With It? 7
  - What is Enterprise Architect? 5
  - Working With 5
  - Workspace Status Bar 117
- Enterprise Architect Lite
  - Availability 18
  - Description 18
  - Edition 18
- Enterprise Architect Simulation Library
  - Behavioral Model Templates 1137
  - EASL Code Generation 1137
  - EASL\_GET Macro 1137
  - EASLList Macro 1137
- Enterprise Architect Toolbox
  - Documentation Group 1774
  - MDG Technology Groups 1031
  - SystemC Group 1474
  - Verilog Group 1477
  - VHDL Group 1478
- Enterprise Java Beans 1314
  - MDG Technology For, Enterprise Architect 1036
- Entity
  - Create 1291
  - Element 1290
  - Element (Entity Relationship Diagram) 1382
- Entity Relationship Diagram
  - Concepts 1382
  - Connector 1382
  - Elements 1382
  - Example Diagram 1382
  - MDG Technology 1382
  - Relationship 1382
  - Tagged Values 1382
  - Toolbox Page 1382
  - Transformation From Data Model 1310
  - Transformation To Data Model 1317
- Entry Point
  - Element 891
- Entry Points
  - Tab Of Structural Specification 669, 675
- Enumeration
  - Automation Interface 1831
  - ConstLayoutStyles 1832
  - CreateBaselineFlag 1832
  - CreateModelType 1832
  - DocumentBreak 1833
  - DocumentType 1833
  - EAEditionTypes 1833
  - Element 952
  - EnumRelationSetType 1834
  - ExportPackageXMIFlag 1834
  - Literal 952
  - MDGMenus 1834
  - MessageFlag 1835
  - ObjectType 1835
  - PropType 1836
  - ReloadType 1837
  - ScenarioDiagramType 1837
  - ScenarioStepType 1838
  - ScenarioTestType 1838
  - XMIType 1838
- Enumeration Elements
  - Add To Profiles 1047
- EnumRelationSetType Enum
  - Automation Interface 1834
- Environment Complexity Factor
  - Definition 402
  - Estimate Project Size 404
  - Estimation 402
  - Value 402
  - Weighting 402
- ERD To Data Model
  - Transformation, MDA-Style Transform 1317
- ERDs 1382
- Eriksson-Penker
  - Concept 1273
  - Diagram 1273
  - Disable 1273
  - Elements 1273
  - Enable 1273
  - Extensions 1273
  - MDG Technology 1273
  - Relationships 1273
  - Toolbox Page 1273
- Eriksson-Penker Business Extensions 1189
- Establish Port-Trigger Mappng
  - Model State Machine For HDL 1502
- Estimation 787
  - Default Hours 404
  - Environment Complexity Factors 402

- Estimation 787
  - Of Project Factors 400
  - Of Project Size 404
  - Of Project Timescale 400
  - Technical Complexity Factors 401
  - Use Case 400
- Estimation Factors
  - Project Types Menu Option 99
- Event
  - Business Modeling 1197
  - Created In Project Calendar 392
  - Displayed In Project Calendar 392
  - Element 1292
  - Receive 1292
  - Recurring, In Project calendar 392
  - Send 1292
- EventProperties
  - Automation Interface Repository 1844
- EventProperty
  - Automation Interface Repository 1844
- Example Diagram
  - Entity Relationship Diagram 1382
- Examples And Tips
  - Automation Interface 1826
- Exception
  - Element 892
- Exception Path
  - Add To Scenario (Toolbar) 671
- Exception Path, Scenario 667
- Exclamation Mark
  - Blue 214
  - Red 214
- Exclude
  - Package In Generated Report 1731
- Exclusive Edit Lock
  - Automatic 196
  - Disable 196
  - Enable 196
  - Toggle 196
- Exclusive Gateway
  - Data-Based 1246
  - Event-Based 1246
- Execution Analysis
  - Add State Transitions 1653
  - Breakpoints And Markers Window, Record Sequence Diagrams 1646
  - Control Recording, Record Sequence Diagrams 1648
  - Diagram Features, Generate Sequence Diagrams 1652
  - Difference Between Recording Marker And Breakpoint 1642
  - Generate Sequence Diagram 1651, 1652
  - Marker Types, Record Sequence Diagrams 1642
  - Object Workbench, Create Variables 1665
  - Object Workbench, Introduction 1663
  - Object Workbench, Invoke Methods 1666
  - Object Workbench, Overview 1664
  - Object Workbench, Variables 1665
  - Place Markers, Recording Sequence Diagrams 1642
  - Profiler Operation 1661
  - Profiler Report, Load 1662
  - Profiler Report, Save 1662
  - Profiler Report, Save As Resource In Team Review 1662
  - Profiler Toolbar 1659
  - Profiler, Attach To Process 1660
  - Profiler, Getting Started 1659
  - Profiler, Launch 1660
  - Profiler, Overview 1658
  - Profiler, Prerequisites 1658
  - Profiler, Set Options 1661
  - Profiler, Set Sample Intervals 1661
  - Profiler, Start 1660
  - Profiler, Stop 1660
  - Profiler, Supported Platforms 1658
  - Profiler, System Requirements 1658
  - Record Activity For Class 1646
  - Record Activity For Method 1645
  - Record Sequence Diagrams, Introduction 1637
  - Record Sequence Diagrams, Overview 1638
  - Record Sequence Diagrams, Prerequisites 1639
  - Record Sequence Diagrams, Set Up 1639
  - Record Unit Test Results 1670
  - Recording History 1651
  - Recording Markers, Activate, Record Sequence Diagrams 1647
  - Recording Markers, Disable, Record Sequence Diagrams 1647
  - Run Unit Test 1670
  - Save Recording History 1653
  - Sequence Diagrams, Limit Auto Recording 1642
  - Set Up To Capture State Changes 1653
  - State Machine Diagram 1655
  - State Transition Diagram 1655
  - Team Review, Save Profiler Report As Resource 1662
  - Unit Test Script, Create 1669
  - Unit Test, Record Results 1670
  - Unit Testing, Introduction 1669

- Execution Analysis
    - Work With Marker Sets, Record Sequence Diagrams 1648
  - Execution Analyzer 1650
    - Record Sequence Diagrams, Recording Options 1639
  - Execution Environment
    - Element 952
  - Execution Profiler
    - Attach To Process 1660
    - Getting Started 1659
    - Launch 1660
    - Operation 1661
    - Overview 1658
    - Prerequisites 1658
    - Report, Example 1658
    - Report, Load 1662
    - Report, Save 1662
    - Report, Save As Resource In Team Review 1662
    - Set Options 1661
    - Set Sample Intervals 1661
    - Start 1660
    - Stop 1660
    - Supported Platforms 1658
    - System Requirements 1658
    - Team Review, Save Report As Resource 1662
    - Toolbar 1659
  - Exit
    - Menu Option (File Menu) 74
  - Exit Point
    - Element 895
  - Expansion Node
    - Action 868
  - Expansion Region
    - Element 893
  - Export
    - .EAB File 290
    - Code Templates 1147
    - Data 237
    - MOF To XMI 1632
    - Profile, To Disk 1050
    - Reference Data 237
    - Reference Data, Introduction 236
    - RTF Style Template 1734
    - State Machine Table To CSV 828
    - To Rational Rose 319, 320, 325
    - To XMI 320
    - Version Controlled Model Branch 290
    - XMI, Batch 332
  - Export Diagrams
    - To RTF Document 1784
  - ExportPackageXMIFlag Enum
    - Automation Interface 1834
  - Expose Interface
    - Element 953
  - Extend
    - Connector 980
    - Relationship 980
    - Toolbox Connectors 1079
    - Toolbox Elements 1077
  - Extended Elements 1288
  - Extended UML Diagrams 1181
  - Extending Modeling Languages
    - And MDG Technologies 1038
    - With Enterprise Architect 1038
    - With UML Patterns In Enterprise Architect 1038
    - With UML Profiles In Enterprise Architect 1038
    - With UML Stereotypes In Enterprise Architect 1038
  - Extension
    - SysML Requirement 1564
  - Extension Points
    - Use Case 937
  - Extension Stereotypes 1288
  - External
    - Element (Data Flow Diagram) 1184
    - Requirements 665, 1164
  - External File
    - Add Link To Team Review Post 227
  - External Requirements
    - Color Coded 1159
  - External Tools
    - Open 121
    - Pass Parameters To 123
- F -**
- Fan Relations Diagram Layout 617
  - Favorites
    - Add To Resources Window 790
    - Delete From Resources Window 790
    - Drag Objects Into 464, 469
    - Elements, Resources Window 790
    - Folder, Resources Window 790
    - Model Views Folder 464
    - View Properties In Resources Window 790
  - FDD Methodology 1166
  - Feature
    - As Attribute 695
    - As Operation 707
    - Element 1166
  - Feature Driven Design Methodology 1166

- Feature Visibility
  - Attributes 585
  - Customize 585
  - Inherited 585
  - Operations 585
  - Set 585
  - Suppress 585
- Features
  - Connect To 742
  - Disconnect From 742
  - Of Enterprise Architect 10
- Feedback Pages 3
- Field Substitution Macros
  - Access Data from Attributes 1116
  - Access Data from Classes 1116
  - Access Data from Operations 1116
  - Access Data from Packages 1116
  - Access Data from Parameters 1116
  - Conditional Substitution 1116
  - Direct Substitution 1116
- Fields and Conditions
  - In Search 489
- File
  - .EAB 290
  - Element Package, Automation Interface 1887
  - Hyperlink To 1293
  - Menu 74
- File Search
  - Introduction 1454
  - List View 1454
  - Search Window, Debugging 1454
  - Toolbar 1454
  - Tree View 1454
  - Use 1454
- Filter
  - Diagram, Display Of Connectors 491
  - Diagram, Display Of Elements 491
- Filter Bar
  - In Diagram List 458
  - In Model Search 480
  - In Package Browser 458
- Filters
  - Add To Search 487
  - Analyzer Script, Recording Options 1639
  - AND 482
  - Diagram, Display Of Connectors 489
  - Diagram, Display Of Elements 489
  - Dynamic Visual 489
  - Element, RTF Report 1770
  - OR 482
  - Other, RTF Report 1771
  - Search 482
- Find
  - Diagram In Project Browser 83
  - Element In All Diagrams 653
  - Element In Diagrams 85
  - Element In Diagrams, Package Browser 459
  - Element In Project Browser 85, 653
  - Submenu 653
- Find And Replace
  - Access Facility 1412
  - Dialog 1412
  - In Code Editor 1412
  - Metasequences 1412
  - Regular Expressions 1412
  - Tagged Regions 1412
- Find In Diagram
  - Element Context Menu, Project Browser 448
- Find in Project
  - Menu Option (Edit Menu) 74
- Find In Project Browser
  - Search Option 444
- Floating Licence
  - Version of Corporate Edition 14
- Floating Toolbar Buttons
  - Display 432
- Floating Windows 103
- Flow Final
  - Element 897
- Font
  - Set For Element Text 657
- Fonts
  - Set Model Default 429
  - Set User Default 429
- Footer
  - Insert In RTF Template 1754
- Footers
  - Add To RTF Document 1799
- Foreign Key
  - Composite, Create 1364
  - Composite, Representation In Diagram 1364
  - Constraint 1361
  - Create 1361
  - Description 1361
  - Name Template, Define 1365
  - Representation In Diagram 1361
- Foreign Language Translation
  - RTF Report Generation 1773
- Fork
  - Element 898, 900
  - Pseudo-State 898, 900
- Fork/Join
  - Element 898
- Formal Statements 19

- Format
    - Element Appearance From Toolbar 541
    - Toolbar 541
  - Forum
    - Discussion, Now Team Review 216
  - Forward Engineering
    - Initial Code In Operations 711
    - Introduction 1386
  - Forward Slash
    - Derived Symbol 696
  - Forward Synchronization
    - Delete Code From Features In Model 1522
  - Fragment
    - Element 880
  - Frame
    - Combine Windows In 103
    - Remove Windows From 103
    - Tabbed 103
  - Free Sorting
    - On Project Browser 455
  - Full Screen
    - Close 100
    - Display 100
  - Fully Qualified Tagged Value
    - Show In Element Compartment 585
  - Function
    - Macros, Code Template Syntax 1130
    - Step Out Of 1449
  - Function Call
    - Step Into 1449
    - Step Through 1449
- G -**
- Gang Of Four Pattern
    - Concepts 1545
    - Download 1021, 1023
    - MDG Technology 1545
    - Toolbox Page 1545
  - Gap Analysis Matrix
    - Edit Notes 512
    - Gap Elements 510
    - Notes, Edit 512
    - Overview 510
    - Profiles 510
  - Gap Element
    - Create 510
  - Gap Notes
    - Edit 512
  - Garden Of Eden Style 1599
  - Gateway
    - Create 1246
    - Exclusive (XOR) 1246
    - Inclusive (OR) 1246
    - Model In BPEL 1246
    - Parallel (AND) 1246
    - Types 1246
  - General
    - Types, Dialog 782
  - General Options
    - Set For Project 422
  - General Ordering
    - Sequence Diagram Messages 994
  - General Types
    - Project Types Menu Option 99
  - Generalization
    - Sets 750
  - Generalization Link
    - Implement Parent Operations 712
    - Override Parent Operations 712
  - Generalize
    - Connector 981
    - Relationship 981
  - Generate
    - BPEL 1253
    - DDL 1379
    - Global Element For Global ComplexTypes 1598
    - Global Element In XSD 1599
    - HTML Report From Virtual Document 1814
    - Package Source Code, (Tools Menu Option) 91
    - Report On Elements 456, 462
    - Report On Project 456
    - Rich Text Format Report (Legacy) 1792
    - RTF Document from Resources (Enhanced) 1766
    - RTF Document From Resources (Legacy) 1793
    - RTF Report (Legacy) 1792
    - RTF Report From Diagram List 462
    - RTF Report From Package Browser 456
    - RTF Report From Virtual Document 1782
    - WSDL 1621
    - XML Schema For Referenced Packages 1598
    - XML Schema, For Child Packages 1598
    - XSD 1598
  - Generate Code
    - For Business Rule 1217
    - From Activity Diagrams 1496, 1504
    - From Behavioral Models 1496
    - From Interaction Diagrams 1496, 1503
    - From Sequence Diagrams 1496, 1503
    - From State Machine Diagrams 1496, 1497
  - Generate HTML Report Dialog 1804
  - Generate RTF Document

- Generate RTF Document
    - Element Filters Tab 1770
    - Other Filters Tab 1771
    - Project Constants, User Defined 1772
    - Update Links In Word 1801
  - Generate RTF Documentation Dialog
    - General Tab 1732
    - Optimize For Sun Open Office 1732
    - Options 1732
    - Template Editor 1743
    - Templates Tab 1734
    - Word Substitution Tab 1773
  - Generate Sequence Diagram
    - Diagram Features 1652
    - Execution Analysis 1651, 1652
  - Generate Source Code
    - Enterprise Architect 7
    - Overview 1490
  - Get All Latest 278
  - Get Function
    - Create As Attribute Property 700
  - Get Latest 278
  - Get Project Custom Colors
    - For Element 642
  - Getting Started
    - Add License Key 2096
    - In Enterprise Architect 34
    - Installing Enterprise Architect 27
    - License Information 2095
    - License Management 2094
    - Register Enterprise Architect 29
    - Registration Key 2095
    - Start Enterprise Architect 35
    - Upgrade Existing License 2099
    - With MOF 1631
  - Global Element
    - Generate For Global ComplexTypes 1598
    - Generate In XSD 1599
    - Import XSD 1601
  - Global Elements
    - Import 1600
  - Global Risks 355
  - Glossary
    - A 2058
    - Add Item, Glossary Dialog 363
    - B 2060
    - C 2061
    - Create Item From Notes Text 769
    - D 2064
    - Delete Item, Glossary Dialog 363
    - Dialog 363
    - E 2066
    - F 2068
    - Filter List, Glossary Dialog 363
    - G 2069
    - H 2070
    - Hyperlink Term From Notes 769
    - I 2071
    - Insert Item In Text 769
    - J 2073
    - L 2074
    - M 2075
    - Model 362
    - Modify Item, Glossary Dialog 363
    - N 2077
    - O 2078
    - Of Terms 2057
    - P 2079
    - Project 362
    - Q 2082
    - R 2083
    - Report 364
    - Report Output Sample 365
    - S 2085
    - T 2088
    - U 2090
    - V 2092
  - Glossary Detail Dialog
    - Add Item 362
    - Modify Item 362
  - Goal
    - Business Modeling 1198
  - GoF Pattern
    - Concepts 1545
    - Download 1021, 1023
    - MDG Technology 1545
    - Toolbox Page 1545
  - Grid
    - Diagram, Adjust 430
  - Group Lock
    - Identify Owner 215
  - Group Login 203
  - Group Properties Dialog
    - Script 1819
  - Guillemets 1019
- H -
- Hardware Description Languages
    - Model State Machine For 1502
  - HDL
    - Model State Machine For 1502
  - Header

- Header
  - Insert In RTF Template 1754
- Headers
  - Add To RTF Document 1799
- Help
  - Add-In 97
  - Display On Add-Ins 97
  - File Formats 31
  - For Tagged Values 765
  - Index Tab 30
  - Learning Center Topics 72
  - Menu 101
  - Release Date 31
  - Search Tab 30
  - Systems 30
  - Topic, Hyperlink To 1292
  - Version Details 31
- Hidden Submenu
  - Create In Toolbox Profile 1075
- Hide
  - Connectors 750
  - Connectors, All Diagrams 751
  - Connectors, Requirements Element 751
  - Connectors, Single Diagram 751
  - Diagram Caption Bar 78
  - Labels 752
  - Package Contents On Diagram 537
  - Project Browser 441
  - Relationship 666
  - Toolbox 546
- Hide/Show
  - Connector Labels 740
  - Connectors 740
- Hierarchy
  - Of Requirements 1164
  - Requirement, Import Via CSV 1162
- Highlight
  - Context Element 645
- History
  - Deep 902
  - Element 902
  - Shallow 902
- Hooped
  - Connectors 745
- Hotkeys
  - Diagram Navigation And Selection 582
- Hourly Rate 404
- HTML
  - And CSS Style Editor 1804
  - Documentation 1724, 1802
  - Generate HTML Report Dialog 1804
  - Locate Page By GUID 1802
  - Quick Start, Generate Report 1802
  - Report 1724, 1802
  - Report Templates 1804
  - Report, Create 1802
  - Style Template Fragments 1806
  - Style Templates, Reporting 1804
  - Template Fragments 1806
  - View Report 1802
- HTML Report
  - Exclude Package 1731
  - Generate From Virtual Document 1814
  - Include Package 1731
- Hyperlink
  - Action Element 1294
  - As Sub Activities 1294
  - Diagrams 1294
  - Element 1292
  - In Linked Document 732
  - Insert In RTF Template 1755
  - Insert New From Toolbar 110
  - To Element From Linked Document 732
  - To External Files 1293
  - To Help Topics 1292
  - To Internet Facilities 1292
  - To Matrix Profiles 1292
  - To Model Search 1292
  - To Script 1294
  - To Team Review 1292
- | -
- Icon
  - Attribute Private 454
  - Attribute Protected 454
  - Checked In Package 454
  - Checked Out Package 454
  - Namespace Root Package 454
  - Operation Private 454
  - Operation Protected 454
  - Project Browser, User Defined 1051
  - Version Controlled Package 454
- ICONIX
  - Disable 1542
  - Elements 1542
  - Enable 1542
  - Layout 1542
  - MDG Technology 1542
  - Relationships 1542
  - Roadmap 1542
  - UML Toolbox Pages 1542
- Icons

- Icons
  - For Element On Diagram 659
  - For Toolbox Items, Assign 1076
- IDE
  - And Enterprise Architect 7
- Image
  - Add To MDG Technology 1071
  - Change Linked To Embedded 1796
  - Element, Add To MDG Technology 1088
  - Handling, RTF 1796
  - Manager 593
  - Print Scaled 602
  - Refresh 593
  - Scale To Page Size 602
  - Select Alternative 593
  - Store In Enterprise Architect 593
  - Use From Image Library 595
- Image Element 1295
- Image Library
  - Import 595
  - Use 595
- Image Manager
  - Create Custom Diagram Background 595
  - Select Alternative Image For Element 593
  - Use 593
- IME
  - Code Editor 1404
- Implementation Manager
  - And Enterprise Architect 65
  - Project Role 65
- Implementation Report
  - Generate 1785
  - In Traceability 494
  - Target Types 1785
- Implemented Interfaces
  - Generate/Disable Methods For 1522
- Implements
  - Connector 1007
  - Relationship 1007
- Import
  - .EAB File 291
  - ActionScript, Reverse Engineering 1510
  - Binary Module (Tools Menu Option) 91
  - Binary Module, Reverse Engineering 1513
  - C#, Reverse Engineering 1510
  - C, Reverse Engineering 1510
  - C++, Reverse Engineering 1510
  - Code Files (Tools Menu Option) 91
  - Code Templates 1147
  - Code, Select Language 109
  - Component Types 1518
  - Data 239
  - Database Schema from ODBC 1375
  - DDL Schema from ODBC 1375
  - Delphi, Reverse Engineering 1510
  - Directory Structure, Reverse Engineering 1513
  - EMX files 324
  - From XMI 323
  - Global Elements 1600
  - Handle Classes Not Found 1515
  - Image Library 595
  - Java, Reverse Engineering 1510
  - MDA-Style Transformations 1331
  - MDG Technologies To APPDATA 1035
  - MOF From XMI 1632
  - Pattern 1023
  - PHP, Reverse Engineering 1510
  - Python, Reverse Engineering 1510
  - Reference Data, Automatically 239
  - Reference Data, Introduction 236
  - Reference Data, Manually 239
  - Referenced XML Schema 1600
  - Requirements Via CSV 1162
  - RTF Style Template 1734
  - RTF Template 1765
  - Scenario as Test 1703
  - Scenarios From Package 1703
  - Source Code, Reverse Engineering 1510, 1515
  - Source Directory (Tools Menu Option) 91
  - Split WSDL Files 1622
  - Technology Files (Option) 90
  - Test From Other Element 1705
  - UML Pattern 1023
  - UML Profiles 1027
  - UML2 files 324
  - User ID From Active Directory 199
  - Version Controlled Model Branch 291
  - Visual Basic, Reverse Engineering 1510
  - Visual Basic.Net, Reverse Engineering 1510
  - WSDL Files 1622
  - XMI 333
  - XSD 1600
- Import/Export
  - Submenu (Project Menu) 81
- Import/Export Submenu
  - Package Context Menu, Project Browser 448
- Imported Class Elements 1377
- in the JBOSS server 1428
- in the Options dialog 1540
- In the Project Browser, right-click on the package into which to paste the copied elements, and select 634
- in the Scripts tab of the Output win 1407
- Inbuilt Stereotypes 1288
- Include

- Include
  - Connector 982
  - Package In Generated Report 1731
  - Relationship 982
- including AspectJ extensions 1470
- Increase Text Display Size 102
- Index
  - Create in Data Modeling 1366
  - Unique 1366
  - What Is An? 1366
- Indicator
  - Locked Element 214
  - Visibility 574
- Information Flow
  - And Patterns 983
  - Connector 983
  - In Combination 983
  - Realized 985
  - Relationship 983
- Information Item
  - Conveyed 984
  - Element 954
  - Find In Project Browser 737, 983
- Inheritance
  - Connector 981
  - Relationship 981
- Inherited Attributes
  - Display 702
- Inherited Feature
  - Show 585
- Inherited Operation
  - Display 713
- Inherited Port
  - Manage 963
- Initial
  - Element 903
- Initial Code
  - Operations Properties 711
- Inline Features
  - Submenu (Element Menu) 87
- Inline IME
  - Option In RTF Editor 1759
- Inline Sequence Elements
  - Part And Port 858
- InnoDB
  - BaseModel Script 172
- In-place Editor
  - Automatic Scroll Into View 720
  - Connector Options 753
  - Edit Attribute Keyword 724
  - Edit Attribute Name 722
  - Edit Attribute Scope 723
  - Edit Attribute Stereotype 722
  - Edit Connector Labels 753
  - Edit Element Name 722
  - Edit Operation Name 722
  - Edit Operation Parameter Keyword 724
  - Edit Operation Parameter Kind 725
  - Edit Operation Scope 723
  - Edit Operation Stereotype 722
  - Element Item, Tasks 720
  - Element Options 720
  - Insert Attribute To Element 726
  - Insert Maintenance Feature 727
  - Insert Operation Parameter 726
  - Insert Operation To Element 726
  - Insert Testing Feature 728
- In-Place Formatting
  - Elements 659
- Input Method Editor
  - Code Editor 1404
- Insert
  - Attribute To Element, In-place Editor 726
  - Bookmarked RTF Into Word 1796
  - Boundary Element 932
  - Diagram Properties Note 587
  - Linked Elements 652
  - Maintenance Feature In Element 727
  - Operation To Element, In-place Editor 726
  - Related Elements 652
  - Testing Features In Element 728
- Install
  - Enterprise Architect 27
- Instance
  - Define Behavior On Creation, Supported Attributes 1055
  - Paste Object As 576
- Instantiated Template 943
- Integrated Development Environment 1390
- Integration Testing
  - Display Details 1698
- Integrity
  - Of Model Data 414
  - Of Project Data 414
- Integrity Check 414
- Intellisense
  - Code Editor, Common 1399
  - In Script Editor 1407
  - Keystrokes, Script Editor 1407
- Interaction
  - Behavioral Aspects 715
  - Diagram 830, 859, 861
  - Edit Parameters 715
  - Element 904

- Interaction
  - Elements and Connectors 556
  - Group, Toolbox 556
  - Parameter 715
  - Parameters, Define 717
- Interaction Diagram
  - Description 849
  - Diagram 849
  - Elements And Connectors 849
  - Example 851
  - Generate Code From 1496, 1503
- Interaction Occurrence
  - Behavior Call 715
  - Element 906
- Interaction Operator
  - Combined Fragment 882
- Interaction Overview Diagram
  - Description 861
  - Elements And Connectors 861
  - Example 862
- InteractionUse
  - Element 906
- Interface
  - Element 955
  - Expose Element 953
  - Provided 953
  - Required 953
  - Set For Element 628
  - Source Code Generation 1490
- Intermediary Language
  - MDA-Style Transforms 1333
- Intermediate Event, BPEL
  - Create 1242
  - Model 1242
  - Types 1242
- Internal Binding
  - PIM to PSM 1301
- Internal Editor
  - Language Properties 1519
- Internal Mail
  - Context Menu 378
  - Create Message 381
  - Delete Messages 378
  - Forward Message 381
  - Hyperlinks 381
  - Internal Mail 378
  - Model Mail Tab 378
  - Options 378
  - Personal Information 381
  - Personal Information Window 378
  - Reply To Message 381
  - Sent Mail Tab 378
  - Toolbar 378
- Internal Requirement
  - Define 1167
  - Properties 1167
- Internet Browser Applets
  - Java, Debug 1424
- Internet Facilities
  - Hyperlink To 1292
- Internet Search Engine
  - Access Within Enterprise Architect 128
  - Define Default 422
- Interrupt Flow
  - Connector 985
  - Relationship 985
- Interruptible Activity Region
  - Element 907
- Interval Bar
  - Context Menu 843
  - Timing Diagram Time Interval 843
- Introduction
  - Copyright Notice 19
  - Enterprise Architect Key Features 10
  - Help File Formats 31
  - License Agreement 20
  - Order Enterprise Architect 26
  - Support 31
  - To Code Engineering 1386
  - To Enterprise Architect 3, 5
  - To Enterprise Architect SDK 1038
  - To Forward Engineering 1386
  - To Quick Linker 622, 1058
  - To Reverse Engineering 1386
  - To Round-trip Engineering 1386
  - To Shape Scripts 1089
  - To Synchronization 1386
  - To Tagged Value Types 1108
  - To UML Objects 864
  - Trademarks 23
- Invocation
  - Associate With Behaviors 716
  - Re-associate With Behaviors 716
  - Synchronize Arguments With Behavior Parameters 716
- Invocation Actions 1504
- Invocations
  - Modeling 715
- Invoke
  - Connector 553
  - Method, Object Workbench 1666
- Issue
  - Create From Test Item 1707
  - Elements And Requirements 1171

Issue (Defect)  
 Add 1719  
 Automation Interface, Element Package 1888  
 Element 1719  
 Hide Stereotype Letter 1719  
 Show Stereotype Letter 1719

Issue (Project)  
 Add 358  
 Delete 358  
 Model 358  
 Modify 358  
 Report, Generate 358

Issues Report  
 Sample Output 361  
 Via Project Issues Tab 360

isUnique  
 UML Property 758

Iterate Through EAP File  
 Automation Interface Code Example 1953

## - J -

Java 1663  
 Advanced Debug Techniques 1423  
 Applets In Internet Browsers, Debug 1424  
 AspectJ Extensions 1510  
 Code Generated From State Machine Diagram 1498  
 Code Generation 1535  
 Debug 1422  
 Debug Session, Attach To VM 1424  
 Debug, System Requirements 1417  
 General Debug Setup 1422  
 Import, Reverse Engineering 1510  
 Language Options 1535  
 Modeling Conventions 1470  
 Modeling Conventions, AspectJ Extensions 1472  
 Transformation, MDA-Style Transform 1319  
 Web Servers, Debugging 1425

JavaScript 1817, 1818

JBOSS  
 Server Configuration 1427  
 Server, Debugging 1425

Jet 3.5  
 Switch To Jet 4.0 35, 422

Jet 4.0  
 Supporting Spell Checking Languages 373  
 Switch From Jet 3.5 35, 422

Join  
 Element 898, 901  
 Pseudo-State 898, 901

JScript 1817, 1818  
 Junction  
 Element 908  
 JUnit Transformation  
 MDA-Style Transform 1321

## - K -

Key  
 Combinations 128  
 Missing Combination 118  
 Key Bindings  
 Code Editor, Common 1400  
 Key Features  
 Of Enterprise Architect 10  
 Key Store  
 File Based 2096  
 Network Based 2096  
 Keyboard  
 Accelerator Map 128  
 And Mouse Shortcuts 133  
 Missing Combination 118  
 Shortcuts 128  
 Shortcuts, Customize 124  
 Shortcuts, Reset 124  
 Key-Mouse  
 Combinations 133  
 Keys  
 Foreign, Definition 1276  
 Primary, Definition 1276  
 Keystore  
 Troubleshooting 2098  
 Keywords  
 For RTF Report (Legacy) 1794

## - L -

Label  
 Alignment 597  
 Bold 597  
 Connector 597  
 Connector, Hide/Show 740  
 Direction Indicator 597  
 Display Under Toolbar Icons 119  
 Element 597  
 Hide 597, 752  
 Menu 597  
 Set Color 597  
 Show 752  
 Show And Hide In Toolbox 549  
 Visibility 752

- Label Visibility
  - On Sequence Messages 858
- Language
  - Adjust in RTF (Legacy) 1794
  - Adjust in RTF Report 1773
  - Create C# Automatic Properties 701
  - Create Properties As Attributes 700
  - Custom, Create Templates For In Code Template Editor 1150
  - For Spell Check 373
  - Macros 1525
  - Other Than English, Spell Checking 373
- Language Options
  - ActionScript 1530
  - Ada 2005 1531
  - C 1531
  - C# 1532
  - C++ 1533
  - Code Editor 1520
  - Code Generation 1529
  - Delphi 1533
  - Java 1535
  - MDG Technology 1540
  - PHP 1536
  - Python 1536
  - SystemC 1537
  - VB.NET 1537
  - Verilog 1538
  - VHDL 1538
  - Visual Basic 1539
- Layout
  - Diagram, Automatically 619
  - Diagram, Move Sections 600
  - ICONIX 1542
  - Sequence Diagram 852
- Layout Diagram
  - Auto Route Layout 618
  - Box Layout 609
  - Center Focussed Layout 605
  - Circular Layout 605
  - Converge Layout 615
  - Digraph Layout 612
  - Diverge Layout 615
  - Elliptical Layout 605
  - Fan Relations Layout 617
  - Main Procedure 604
  - Neaten Layout 614
  - Options 604
  - Per Page Layout 611
  - Spring Layout 613
  - Tool 604
  - Top-To-Bottom Circle 605
- Layout Tools Window 604
- Lazy Loading
  - Enterprise Architect Performance 162, 163, 164, 166, 167, 169, 170
  - For ASA Data Repository 169
  - For MSDE Server Data Repository 170
  - For MySQL Data Repository 162
  - For Oracle Data Repository 164, 166
  - For PostgreSQL Data Repository 167
  - For Progress OpenEdge Data Repository 170
  - For SQL Server Data Repository 163
- Learning Center
  - Add To MDG Technology 1069
  - Allocate Toolbox To Contexts 1086
  - Commands, Built In 1084
  - Contexts, Define 1085
  - Getting Started 72
  - Introduction 30
  - Named Contexts 1085
  - Run Add-In Functions From 1084
  - Save Profile 1086
  - Tools 72
  - Window 72
- Learningd Center
  - Profiles, Create 1082
- Leasrning Center
  - Toolboxes, Define 1083
- Legend
  - Add To Diagram 588
  - Add To State Machine Table 826
  - Edit 588
  - Element 588
  - Ignore 826
  - Insert New From Toolbar 110
  - Never Happen 826
  - Remove From State Machine Table 826
  - Style 588
- Letter A
  - Red 729, 730
- Level Numbering
  - For Requirements 1164
  - Show In Project Browser 444
- License
  - Agreement 20
  - Information 2095
  - Management 2094
- Lifecycle
  - Of A Sequence Element 851
- Lifeline
  - Element 910
  - Objects In Sequence Diagrams 854
  - Sequence Element, Termination 851

- Limit Auto Recording to Stack Frame Threshold
  - Marker Types, Record Sequence Diagrams 1642
- Limitations
  - Of XMI 325
- Line
  - Angles, Tidy 745
  - Bend At Cursor 745
  - Bezier 745
  - Straighten At Cursor 745
  - Style 745
  - Suppress Segments 745
- Line Points
  - Add 745
  - Delete 745
  - Toggle 745
- Line Selection
  - Code Editor, Common 1399
- Link
  - Add Note 743
  - Copy Between Instances Of Elements From Project Browser 748
  - Create Between Elements 748
  - Create From Project Browser 748
  - Display Options 434
  - Element Feature To Note 640
  - Note To Element Feature 640
  - Paste Object As 576
  - Window 504
- Link Notes
  - To Element Feature 640
  - To Internal Documentation 640
- Linked Document
  - Create 729
  - Create Diagram From 732
  - Create Document Artifact 730
  - Create Element From 732
  - Create For UML Element 730
  - Create Glossary Definition In 731
  - Delete 733
  - Edit 731
  - Editor Context Menu 731
  - Hyperlink 732
  - Introduction 729
  - Link Into RTF Report 729, 1737
  - Replace 733
  - To UML Element 730
- Linked Document Template
  - Add To MDG Technology 1072
  - Assign To Group 734
  - Create 734
  - Delete 734
- Edit 734
- Modify 734
- Linked Element
  - Convert To Local Copy 646
- Linked Image
  - Change To Embedded 1796
- List Macro
  - Code Template Syntax 1133
- List Overrides
  - RTF Reports 1761
- Load
  - Controlled Package 332
  - Report Templates (Legacy) 1792
- Loaded Modules
  - Show In Debugger 1446
- Local
  - Directories 1524
  - Path Dialog 1524
  - Paths 1523
  - Pre/Post Conditions 871
- Local Variables
  - View 1442
  - View Long Values 1443
- Locals Window 1442
  - View Long Values 1443
- Lock
  - Apply User Lock 213
  - Connector 207
  - Delete 207
  - Delete, User level 215
  - Diagram, Rigorous Security Mode 213
  - Diagram, Security Off 602
  - Diagram, User/Group Lock 210
  - Element 207
  - Element, Security Off 655
  - Identify Owner 215
  - Manage 207
  - Manage, User-Level 215
  - Model Elements, Rigorous Security Mode 213
  - Model Elements, User/Group Lock 210
  - Package, Rigorous Security Mode 213
  - Package, User/Group Lock 210
  - Packages 212
  - Release User Lock 213
  - Standard Security Policy 210
  - View 207
  - View, User Level 215
- Locked Element
  - Add Connectors 211
  - Indicators 214
- Log
  - Audit, Clear 300

- Log
  - Audit, Load 300
  - Audit, Save 300
- Logical Diagram
  - Class Diagram 798
- Logical Model 523
- Login
  - As Different User (Security) 82
  - Group 203
  - Multiple Under One ID 203
- Loop Node
  - Structured Activity 926, 928
  
- M -**
  
- Macro
  - Behavioral Model 1137
  - Branching 1133
  - Code Template Syntax 1115, 1133
  - Control 1133
  - CONVERT\_DB\_TYPE 1340
  - CONVERT\_NAMES 1341
  - CONVERT\_TYPE 1340
  - EASL Code Generation 1137
  - EASL\_GET 1137
  - EASList 1137
  - Field Substitution, Code Template Syntax 1116
  - Function, Code Template Syntax 1130
  - Language 1525
  - List 1133
  - PI 1133
  - Preprocessor 1525
  - REMOVE\_PREFIX 1341
  - SQL Search 484
  - Synchronization 1133
  - Tagged Value, Code Template Syntax 1129
  - Template Substitution, Code Template Syntax 1116
  - TRANSFORM\_CLASSIFIER 1342
  - TRANSFORM\_CURRENT 1340
  - TRANSFORM\_REFERENCE 1338, 1342
- Mail Interface Package
  - Automation Interface 1950
  - MailInterface, Automation Interface 1950
- MailInterface Class
  - Automation Interface, Mail Interface Package 1950
- Main Context Menu 73
- Main Menu 73
- Maintain
  - Groups 203
  - Security Users 198
- Maintenance 1715
  - Asterisk On Maintenance Window Tabs 1713
  - Compartment, Element 659
  - Create Defect Item From Test 1707
  - Dialog 787
  - Elements And Connectors 564
  - Feature, Insert In Element 727
  - Group, Toolbox 564
  - Items 1713
  - Model Template 528
  - Of Element Properties 1715
  - Problem Types 787
  - Script, Show In Compartments 1717
  - Support 1715
  - Testing Types 788
  - Window 1713
  - Workspace 1713
- Maintenance Data
  - Project Types Menu Option 99
- Maintenance Diagram
  - Description 1711
  - Elements And Connectors 1711
  - Example 1712
- Maintenance Item
  - Copy Between Categories 1716
  - Create Element From 1717
  - Move Between Categories 1716
- Make Same
  - Submenu 88
- Manage
  - Add-Ins 1974
  - Baselines 311
  - Bookmarks 412
  - Changes 241
  - Diagrams 567
  - Elements 625
  - Inherited Ports 963
  - Locks 207
  - MDG Technologies 1033
  - Redefined Ports 963
  - Requirements 1169
  - User-Level Locks 215
  - Views 530
  - Views, Add Views 531
  - Views, Delete 532
  - Views, Rename 531
- Manage .EAP File
  - Submenu (Data Management Menu) 94
- Manage Uses Context List Dialog 674
- Managed C++
  - Modeling Conventions 1467
- Manifest

- Manifest
  - Connector 986
  - Relationship 986
- Manual Version Control
  - With XMI 334
- Map State Changes
  - Visual Execution Analyzer 1657
- Mapper
  - Data Type Conversion Procedures 1356
- Marker
  - Sets 1646
  - Storage 1441
  - Types 1642, 1646
- Marker Management
  - Debugger 1436
- Marker Sets
  - Work With 1648
- Masked Tagged Values
  - Create 1113
- Master Document
  - Element 1774
  - Element, Create 1776
  - Generate Documentation 1782
  - Generate HTML Documentation 1814
  - Overview, Virtual Document 1774
  - RTF Bookmarks In 1774
  - Tagged Values 1776
- Master Document (HTML) 1813
- Matrix
  - Swimlanes 592
- Matrix Profile
  - Hyperlink To 1292
- MDA Transform
  - Add To MDG Technology 1071
- MDA Transformation
  - Built-In 1301
  - Enterprise Architect 7
  - Overview 1301
  - Tools Menu Option 93
- MDA-Style Transformation
  - Built In Transformation 1307
  - C# Transformation 1307
  - Chaining Transformations 1306
  - Convert Datatypes 1340
  - Convert Names 1341
  - Convert Types 1340
  - Copy Information 1340
  - Create Connectors 1338
  - Cross References 1342
  - Data Model To ERD Transformation 1310
  - DDL Transformation 1310
  - Duplication Of Connectors 1338
  - EJB Transformations 1314
  - ERD To Data Model Transformation 1317
  - Import Transformations 1331
  - Intermediary Language 1333
  - Java Transformation 1319
  - JUnit Transformation 1321
  - NUnit Transformation 1323
  - Remove Prefix 1341
  - Specify Classifiers 1342
  - Transform Connectors 1338
  - Transform Elements 1305
  - Transformation Templates 1330
  - Write Transformations 1332
  - WSDL Transformation 1325
  - XSD Transformation 1326
- MDDE
  - Advanced Debug Techniques, Java 1423
  - Analyzer Scripts, Manage 1392
  - Available Tools 1391
  - Basic Setup 1392
  - Breakpoint Management 1436
  - Build Script, Create 1415
  - Build Script, Introduction 1415
  - Debug .NET 1428
  - Debug .NET CLR Versions 1429
  - Debug .NET With COM Interop Process 1430
  - Debug Apache Tomcat Server Configuration 1428
  - Debug Apache Tomcat Windows Service 1428
  - Debug ASP .NET 1431
  - Debug Java 1422
  - Debug Java Applets In Internet Browsers 1424
  - Debug Java Web Servers 1425
  - Debug JBOSS Server Configuration 1427
  - Debug Symbols, C++ And Native Applications 1421
  - Debugger Frameworks 1416
  - Debugger System Requirements 1417
  - Debugger, Overview 1416
  - External Tools 1392
  - For C++ Applications 1420
  - For Microsoft Native Applications 1420
  - For WINE Applications 1419
  - General Debug Setup, Java 1422
  - General Workflow 1391
  - Generate Code 1392
  - Getting Started 1391
  - Introduction 1390
  - Java Debug Session, Attach To VM 1424
  - Limitations 1387
  - Marker Management 1436
  - Overview 1387

## MDDE

Prerequisites 1391  
 Recursive Builds 1416  
 Script Actions, Define 1394  
 Set Up Debug Session 1417  
 Set Up Debug Session For .NET 1429  
 Supported Environments 1387  
 Synchronize Code 1392  
 UAC-Enabled Operating Systems 1418  
 Workspace Layout 1391

## MDG Add-In For

CORBA 1036  
 Data Distribution Service 1036  
 DDS 1036  
 Department Of Defense Architecture Framework -  
 Ministry Of Defence Architecture Framework  
 1036  
 DoDAF-MODAF 1036  
 DOORS 1036  
 Eclipse 1036

## MDG Add-Ins

Add-In Model 2042  
 MDG Events 2042  
 MDG\_BuildProject 2043  
 MDG\_Connect 2044  
 MDG\_Disconnect 2045  
 MDG\_GetConnectedPackages 2046  
 MDG\_GetProperty 2046  
 MDG\_Merge 2047  
 MDG\_NewClass 2049  
 MDG\_PostGenerate 2050  
 MDG\_PostMerge 2051  
 MDG\_PreGenerate 2051  
 MDG\_PreMerge 2052  
 MDG\_PreReverse 2053  
 MDG\_Run\_Exe 2054  
 MDG\_View 2054

## MDG Events

Add-In Model 2042  
 MDG\_BuildProject 2043  
 MDG\_Connect 2044  
 MDG\_Disconnect 2045  
 MDG\_GetConnectedPackages 2046  
 MDG\_GetProperty 2046  
 MDG\_Merge 2047  
 MDG\_NewClass 2049  
 MDG\_PostGenerate 2050  
 MDG\_PostMerge 2051  
 MDG\_PreGenerate 2051  
 MDG\_PreMerge 2052  
 MDG\_PreReverse 2053  
 MDG\_Run\_Exe 2054

MDG\_View 2054

## MDG Integration For

Eclipse 1036  
 TcSE 1036  
 Teamcenter Systems Engineering, Siemens PLM  
 1036  
 Visual Studio 2005/2008 1036

## MDG Link For

DOORS, Telelogic 1036  
 Eclipse 1036  
 Visio 1036  
 Visual Studio.NET 1036

## MDG Technology

Access, Remote From Enterprise Architect  
 1034  
 Activate 1033  
 Active, Automatically Pinned Pages 549  
 And Resources Window 1031  
 Code Generation 1540  
 Create 1064  
 Create Toolbox Profile For 1074  
 Define Learning Center Profile 1082  
 Define Validation Configuration 1086  
 Deploy From Add-In 1089  
 Deploy From File 1089  
 Disable 1033  
 Enable 1033  
 Example Of Development 1064  
 Example Of Element Image Setup 1088  
 Import To APPDATA 1035  
 Include Custom Diagram Types 1080  
 Incorporate Model Template 1087  
 Introduction 1064  
 Introduction, Technology Sources 1031  
 Language Options 1540  
 Link for Downloads 1031  
 Manage 1033  
 Toolbox Groups 1031  
 Win32@ User Interface 1286

## MDG Technology (Integrated)

ArchiMate 1272  
 BPEL 1235  
 BPMN 1219  
 Data Flow Diagram 1184  
 Entity Relationship Diagram 1382  
 Eriksson-Penker 1273  
 Gang Of Four Pattern 1545  
 GoF Pattern 1545  
 ICONIX 1542  
 Mind Mapping 1182  
 SoaML 1624  
 SOMF 1628

- MDG Technology (Integrated)
  - SPEM 406
  - SysML 1552
- MDG Technology For
  - CORBA 1036
  - Data Distribution Service 1036
  - DDS 1036
  - Department Of Defense Architecture Framework - Ministry Of Defence Architecture Framework 1036
  - DoDAF-MODAF 1036
  - Enterprise Java Beans 1036
  - Python 1036
  - SysML 1036
  - Systems Modeling Languages 1036
  - Testing 1036
  - The Open Group Architecture Framework 1036
  - TOGAF 1036
  - Zachman Framework 1036
- MDG Technology Selection (MTS) File 1073
- MDG Technology Wizard
  - Add Code Modules 1070
  - Add Diagram Type To Technology 1068
  - Add Images 1071
  - Add Learning Center To Technology 1069
  - Add Linked Document Template To Technology 1072
  - Add MDA Transforms 1071
  - Add Pattern To Technology 1067
  - Add Profile To Technology 1066
  - Add RTF Report Template To Technology 1072
  - Add Scripts 1072
  - Add Tagged Value Types 1069
  - Add Toolbox To Technology 1068
  - Create Technologies 1064
- MDG\_BuildProject
  - Add-In Model 2043
- MDG\_Connect
  - Add-In Model 2044
- MDG\_Disconnect
  - Add-In Model 2045
- MDG\_GetConnectedPackages
  - Add-In Model 2046
- MDG\_GetProperty
  - Add-In Model 2046
- MDG\_Merge
  - Add-In Model 2047
- MDG\_NewClass
  - Add-In Model 2049
- MDG\_PostGenerate
  - Add-In Model 2050
- MDG\_PostMerge
  - Add-In Model 2051
- MDG\_PreGenerate
  - Add-In Model 2051
- MDG\_PreMerge
  - Add-In Model 2052
- MDG\_PreReverse
  - Add-In Model 2053
- MDG\_Run\_Exe
  - Add-In Model 2054
- MDG\_View
  - Add-In Model 2054
- MDGMenu Enum
  - Automation Interface 1834
- Memory Viewer
  - Window 1445
- Menu
  - Add-Ins 97
  - Analyzer 94
  - Context 69
  - Context, Project Browser 443
  - Customize Appearance 125
  - Delete Options 125
  - Diagram 83
  - Diagram Context 538
  - Diagram, Advanced 84
  - Edit 74
  - Element 85
  - Element Context 648
  - File 74
  - Help 101
  - Items, Define In Add-In 1969
  - Label 597
  - Main 69, 73
  - Main Context 73
  - Missing Option 118
  - More Element Tools 76
  - More Project Tools 76
  - New Element Or Connector Context 540
  - Package Control 328
  - Project 80
  - Properties, Element Context 649
  - Set Animation 125
  - Set Shadowing 125
  - Settings (Project) 98
  - Toolbars (View Menu Option) 76
  - Tools 90
  - Version Control (Package) 278
  - Version Control (Project) 277
  - View 76
  - Visual Styles (View Menu Option) 76
  - Window 100
- Menu Option (File Menu)

- Menu Option (File Menu)
  - Close Project 74
  - Exit 74
  - New Project 74
  - Open Project 74
  - Page Setup 74
  - Print 74
  - Print Preview 74
  - Print Setup 74
  - Reload Current Project 74
  - Save Project As 74
- Merge
  - Baseline With Current Model, Overview 308
  - Element 886, 910
  - Node 910
  - Options, Compare Utility 316
- Merge Packages
  - Relationship 1006
- Message 754
  - Asynchronous Signal 996
  - Collaboration 998
  - Colors In Communication Diagrams 861
  - Communication 996
  - Communication, Coloring 435
  - Communication, Create 997
  - Connector 986
  - Create On Timing Diagram 1000
  - Endpoint 911
  - Group, Start New 998
  - Label 912
  - Level 998
  - Move 986
  - Recursion 1008
  - Relationship 986
  - Re-Order 998
  - Scope 754
  - Self Message 989
  - Self Message Call 991
  - Sequence Communication 998
  - Sequence Diagram, Asynchronous Signal 996
  - Sequence Diagram, Examples 992
  - Sequence Diagram, General Ordering 994
  - Sequence Diagram, Self Message 989
  - Sequence, Create 987
  - Sequence, Label Visibility 858
  - Sequencing 998
  - Source and Target 754
  - Timing Diagram 999
  - WSDL Diagram 1613
  - WSDL Element 1613
- Message Angle
  - Adjust With Duration Constraint 993
- Message Part
  - WSDL Attribute 1614
- MessageFlag Enum
  - Automation Interface 1835
- Meta Object Facility
  - Introduction 1630
- Metaclass
  - Add To Profile 1042
- META-INF Package 1314
- Metamodel
  - Elements and Connectors 561
  - Group, Toolbox 561
- Metasequence
  - Code Editor Find And Replace 1412
  - Helper Menu 1412
- Method
  - Add And Delete, Automation Interface Code Example 1957
  - Automation Interface, ElementFeatures Package 1903
  - Context Menu, Project Browser 452
  - Create Use Case For 1406
  - Delete If Not In Code In Reverse Synchronization 1522
  - From Rule Flow Activity 1208
  - Implemented Interfaces 1522
  - Include Bodies In Model When Reverse Engineering 1522
  - Invoke, Object Workbench 1666
  - Link To Use Case 1406
  - Show Parameters On Diagram 575
  - Work With, Automation Interface Code Example 1965
- MethodConstraint
  - Automation Interface, ElementFeatures Package 1905
- Methodology
  - FDD 1166
  - Feature-Driven Design 1166
- MethodTag
  - Automation Interface, ElementFeatures Package 1906
- Metric
  - Automation Interface, Element Package 1889
- Metric Types 354
  - Define 354
  - Global 354
  - Non-Global 352
- Metrics 787
  - Project Management Window 347
- Metrics And Estimation
  - Default Hour Rate 404
  - ECF 402

- Metrics And Estimation
  - Effort Types 353
  - Environment Complexity Factors 402
  - For An Element 352
  - Metric Types 354
  - Risk Types 355
  - TCF 401
  - Technical Complexity Factors 401
- Microsoft .NET 1429
- Microsoft Native
  - Debug Symbols 1421
  - Set Up Debug Sessions 1420
- Microsoft Word
  - Special Considerations 1798
  - Use In RTF Documentation 1795
- Migrate
  - BPMN 1.0 Model To 1.1 1232
  - BPMN 1.1 Model To 2.0 1233
  - SysML 1.1 Model To 1.2 1578
- Migrate()
  - Function, BPMN 1232
  - Function, BPMN 2.0 1233
  - Function, SysML 1578
- MigrateToBPMN11()
  - Function 1232, 1929
- Migration
  - From UML 1.3 414
  - To UML 2.0 414
- Mind Mapping
  - Concept 1182
  - Diagram 1182
  - Disable 1182
  - Elements 1182
  - Enable 1182
  - MDG Technology 1182
  - Relationship 1182
  - Toolbox Page 1182
- MiscData 1877
- Model
  - Activity, BPEL 1244
  - Add To Project 136
  - Analysis, Business Processes 1195
  - Automation Interface 1829
  - BPEL Process 1240
  - BPEL, Create 1238
  - Business Domain, Create 1204
  - Business Processes, Analysis 1195
  - Business Rules 1199
  - Contents 518
  - Context Menu, Project Browser 443
  - Data Integrity 414
  - Databases 1276
  - Default Diagram, Cancel 585
  - Default Diagram, Set 585
  - Default Fonts 429
  - Delete Element From 637
  - Delete Multiple Elements From 637
  - Gateway, BPEL 1246
  - Glossary 362
  - Integrity Check 414
  - Integrity, Run SQL Patches 418
  - Issues, Record 358
  - Navigation Tools 440
  - Package, Create Using Model Wizard 518
  - Pattern 518
  - Pool, BPEL 1249
  - Remove Recent 71
  - Requirements 1152, 1164
  - Requirements, SysML, Create 1572
  - Root Node 443
  - Search Tools 440
  - Sequence Flow, BPEL 1248
  - Sharing, Introduction 185
  - Shortcut 140
  - Shortcut (Capture Current Environment) 142
  - Shortcut (Direct Definition) 141
  - Simulation 1681
  - Structure 515
  - SysML Operational Domain, Create 1573
  - SysML Parametric, Create 1568
  - SysML System Design 1575
  - SysML, Simulate 1570
  - Systems Engineering, Create 1548
  - Templates, Incorporate In Technology 1087
  - Trace Tools 440
  - Transformation 1305
  - UML, Specialized 1178
  - Upgrade 417
  - What Is A? 518
  - WSDL 1612
  - WSDL, Binding 1617
  - WSDL, Document 1620
  - WSDL, Message 1613
  - WSDL, Message Part 1614
  - WSDL, Port Type 1614
  - WSDL, Port Type Operation 1615
  - WSDL, Service 1619
  - XSD 1581
- Model Branch
  - .EAB File 290
  - Apply Version Control 289
  - Export 290
  - File 290
  - Import 291

- Model Changes
  - Auditing 298
  - Record 298
- Model Document
  - Add Packages As Attributes 1778
  - Change Package Sequence 1780
  - Delete Package Attributes 1779
  - Delete Packages 1779
  - Document Order 1780
  - Element 1774
  - Element, Create 1777
  - Generate Documentation 1782
  - Generate HTML Documentation 1814
  - Identify Search For 1777
  - Model Search Sequence 1780
  - Move Package Attributes Between Elements 1780
  - Overview, Virtual Document 1774
  - Select Template For 1777
  - Sequence of Model Document Elements 1780
  - Sequence of Package Attributes 1780
  - Tagged Values 1777
- Model Document (HTML) 1813
- Model Driven Achitecture
  - Overview 1301
- Model Driven Development Environment 1390
  - Advanced Debug Techniques, Java 1423
  - Analyzer Scripts, Manage 1392
  - Available Tools 1391
  - Basic Setup 1392
  - Breakpoint Management 1436
  - Build Script, Create 1415
  - Build Script, Introduction 1415
  - Debug .NET 1428
  - Debug .NET CLR Versions 1429
  - Debug .NET With COM Interop Process 1430
  - Debug Apache Tomcat Server Configuration 1428
  - Debug Apache Tomcat Windows Service 1428
  - Debug ASP .NET 1431
  - Debug Java 1422
  - Debug Java Applets In Internet Browsers 1424
  - Debug Java Web Servers 1425
  - Debug JBOSS Server Configuration 1427
  - Debug Symbols, C++ And Native Applications 1421
  - Debugger Frameworks 1416
  - Debugger System Requirements 1417
  - Debugger, Overview 1416
  - External Tools 1392
  - For C++ Applications 1420
  - For Microsoft Native Applications 1420
  - For WINE Applications 1419
  - General Debug Setup, Java 1422
  - General Workflow 1391
  - Generate Code 1392
  - Getting Started 1391
  - Java Debug Session, Attach To VM 1424
  - Limitations 1387
  - Marker Management 1436
  - Overview 1387
  - Prerequisites 1391
  - Recursive Builds 1416
  - Script Actions, Define 1394
  - Set Up Debug Session 1417
  - Set Up Debug Session For .NET 1429
  - Supported Environments 1387
  - Synchronize Code 1392
  - UAC-Enabled Operating Systems 1418
  - Workspace Layout 1391
- Model Driven Generation
  - Add-Ins 1036
- Model File
  - Connect To ASA Data Repository 169
  - Connect To Data Repository 161
  - Connect To MSDE Server Data Repository 170
  - Connect To MySQL Data Repository 162
  - Connect To Oracle 10g Data Repository, Via ODBC 164
  - Connect To Oracle 10g Data Repository, Via OLE DB 166
  - Connect To Oracle 11g Data Repository, Via ODBC 164
  - Connect To Oracle 11g Data Repository, Via OLE DB 166
  - Connect To Oracle 11g Data Repository, Via OLE DB 166
  - Connect To Oracle 9i Data Repository, Via ODBC 164
  - Connect To Oracle 9i Data Repository, Via OLE DB 166
  - Connect To PostgreSQL Data Repository 167
  - Connect To Progress OpenEdge Data Repository 170
  - Connect To SQL Server Data Repository 163
  - Create Access 2007 Repository 149
  - Create Adaptive Server Anywhere Repository 152
  - Create Data Repository 148
  - Create Model 145
  - Create MSDE Server Repository 152
  - Create MySQL Repository 149
  - Create Oracle 10g Server Repository 151
  - Create Oracle 11g Server Repository 151
  - Create Oracle 9i Server Repository 151
  - Create PostgreSQL Repository 151
  - Create Progress OpenEdge Repository 153

- Model File
  - Create SQL Server Repository 150
  - Enterprise Architect Project Files 136
  - Set Up Adaptive Server Anywhere ODBC Driver 158
  - Set Up Database Model Files 148
  - Set Up MySQL ODBC Driver 154
  - Set Up ODBC Driver 154
  - Set Up Oracle ODBC Driver 160
  - Set Up PostgreSQL ODBC Driver 155
  - Set Up Progress OpenEdge ODBC Driver 159
- Model Glossary
  - Add Item, Glossary Detail Dialog 362
  - Add Item, Glossary Dialog 363
  - Delete Item, Glossary Dialog 363
  - Delete Item, Project Glossary Tab 362
  - Filter List, Glossary Dialog 363
  - Filter List, Project Glossary Tab 362
  - Glossary Dialog 363
  - Glossary Report 364
  - Modify Item, Glossary Detail Dialog 362
  - Modify Item, Glossary Dialog 363
  - Project Glossary Tab 362
  - Redefine Entry Type, Project Glossary Tab 362
- Model Mail
  - Create Message 381
  - Forward Message 381
  - Personal Information Window 378
  - Reply To Message 381
- Model Maintenance (.EAP)
  - Compact Project 418
  - Introduction 414
  - Rename Project 418
  - Repair Project 419
- Model Message
  - Dialog 381
- Model Pattern
  - Introduction 519
  - Select In Enterprise Architect 37
- Model Profile
  - Apply 592
  - Save 592
  - Swimlanes Matrix 592
  - Zachman 592
- Model Search
  - Access From Add-In 1974
  - Access From Shortcut 140, 142
  - Access From Shortcut (Direct Definition) 141
  - Add Data Columns 475
  - Add Filters 487
  - Advanced Search Options 486
  - Conditions 489
  - Create Search Definition 484
  - Customize Search View 480
  - Define In MTS File 1073
  - Export Search Definitions 482
  - Fields 489
  - Filter Bar 480
  - Generate RTF Report 1727
  - Group Results 475
  - Hyperlink To 1292
  - Import Search Definitions 482
  - Introduction 475
  - Manage 482
  - Notes Options 475
  - Options 475
  - Process Results Of Search 475
  - Remove Data Columns 475
  - Sort And Select 480
  - To Populate Model Views 469
  - Toolbar Options 475
  - Value Grouping 480
  - View 475
  - View Header 480
- Model Simulation
  - Activate Simulation Script 1682
  - Breakpoint Sets 1684
  - Breakpoints, Using 1684
  - How It Works 1681
  - Run Simulation Script 1683
  - Set Up Simulation Script 1681
  - Supported Platforms 1681
  - Topics 1681
- Model State Machine
  - Active State Logic 1502
  - Designate Driving Triggers 1502
  - Establish Port-Trigger Mapping 1502
  - For Hardware Description Languages 1502
- Model Template
  - Business Process 520
  - Class 523
  - Component 525
  - Database 524
  - Deployment 526
  - Domain 522
  - Introduction 519
  - Logical 523
  - Maintenance 528
  - Physical 526
  - Project 528
  - Relational Database 524
  - Requirements 521
  - Testing 527
  - Use Case 521

- Model Transfer
  - Introduction 318
- Model Transformations
  - Submenu (Tools Menu) 93
- Model Validation
  - Configure 1689
  - Define Configuration For MDG Technology 1086
  - Element Composition Rule 1691
  - Object Constraint Language 1688
  - OCL 1688
  - OCL Conformance Rule 1692
  - Property Validity Rule 1692
  - Rule, Element Composition 1691
  - Rule, OCL Conformance 1692
  - Rule, Property Validity 1692
  - Rule, Well Formedness 1691
  - Rules 1690
  - Submenu (Project Menu) 81
  - Well Formedness Rule 1691
- Model Validation Broadcasts
  - Add-In Model 1999
  - EA\_OnEndValidation 2001
  - EA\_OnInitializeUserRules 2000
  - EA\_OnRunAttributeRule 2004
  - EA\_OnRunConnectorRule 2003
  - EA\_OnRunDiagramRule 2002
  - EA\_OnRunElementRule 2001
  - EA\_OnRunMethodRule 2005
  - EA\_OnRunPackageRule 2002
  - EA\_OnRunParameterRule 2005
  - EA\_OnStartValidation 2000
  - Model Validation Example 2006
- Model Views
  - Automatic Notification 413
  - Context Menu Options 467
  - Define In MTS File 1073
  - Define Search 469
  - Delete (Context Menu) 467
  - Delete (Toolbar) 466
  - Display Recent Postings 469
  - Enable Technology-Defined View 469
  - Export As XML 469
  - Favorites 464
  - Favorites Folder, Create (Context Menu) 467
  - Favorites Folder, Create (Toolbar) 466
  - For Requirement Change Management 1171
  - Import As XML 469
  - Monitor Work Flow 413
  - Move Objects Between Views 469
  - Move Objects Into Favorites 469
  - My Views 464
  - Operations 469
  - Postings From Team Review 464
  - Properties 469
  - Recent Discussions 464
  - Recent Post Options 467
  - Refresh (Toolbar) 466
  - Refresh Search 469
  - Root Node, Create (Context Menu) 467
  - Root Node, Create (Toolbar) 466
  - Slide Show, Run (Manual) 471
  - Slideshow 464
  - Slideshow Folder, Create (Toolbar) 466, 471
  - Slideshow, Automate 471
  - Technology-Defined 464
  - Toolbar Options 466
  - Use Objects In Views And Favorites 469
  - View, Create (Context Menu) 467
  - View, Create (Toolbar) 466
  - Views 464
  - Views Folder, Create (Context Menu) 467
  - Views Folder, Create (Toolbar) 466
  - Window 464
- Model Violations
  - Examples 1688
- Model Wizard
  - Add Model Package 518
  - Quick Start 37
  - Use To Create Model 145
- Modeling
  - Behavior Calls 715
  - Behaviors (General) 706
  - Enterprise Architect Processes 5
  - Enterprise Architect Tools and Features 69
  - Enterprise Architect, Getting Started 34
  - Getting Started - Enterprise Architect 34
  - Invocations 715
  - Systems Engineering 1548
  - Tasks With Enterprise Architect 5
  - The Business Process 1193
  - With Diagrams 538
  - With Elements 625
  - With Enterprise Architect, Overview 517
- Modeling Conventions 1457
  - ActionScript 2 and 3 1458
  - Ada 2005 1459
  - ANSI C 1461
  - C 1461
  - C# 1463
  - C, Object Oriented Programming 1462
  - C++ 1465
  - C++, Managed 1467
  - C++/CLI Extensions 1468
  - Delphi 1469

- Modeling Conventions 1457
  - Java 1470
  - Java AspectJ Extensions 1472
  - Object Oriented Programming in C 1462
  - PHP 1472
  - Python 1473
  - SystemC 1474
  - VB.NET 1475
  - Verilog 1477
  - VHDL 1478
  - Visual Basic 1480
- Modeling Language
  - Extending 1038
- Modeling Tool
  - Enterprise Architect 3
- Models Collection 1855
- Model-View-Controller Pattern 1290
- ModelWatcher
  - Automation Interface Repository 1845
- Modify
  - Element Changes 1715
  - Element Defects 1715
  - Element Issues 1715
  - Element Tasks 1715
  - Linked Document Template 734
  - Project Task 357
  - Relationship Using Matrix 500
  - RTF Style Template (Legacy) 1792
  - Tagged Values 764
  - Z-Order 538
- Modules
  - Window 1446
- MOF
  - Export To XMI 1632
  - Getting Started 1631
  - Import From XMI 1632
  - Introduction 1630
- Monitor
  - Events 413
  - Progress 413
  - Work Flow 413
- Monitor Project Tasks
  - Personal Information Window 385
- Monitor Workflow
  - Personal Information Window 386
- More Analyzer Windows
  - Memory Viewer 97
  - Modules 97
  - Scripts 97
  - Submenu 97
  - Workbench 97
- More Element Tools
  - Submenu 78
- More Project Tools
  - Submenu 78
- Mouse/Keyboard
  - Shortcuts 133
- Mouseovers
  - Code Editor, Common 1399
- Move
  - Attributes Between Elements 647
  - Connectors 744
  - Connectors, Quick Start 44
  - Diagram Sections 600
  - Diagrams, Quick Start 44
  - Elements Between Packages 633
  - Elements By Increments 88, 600
  - Elements In Diagrams 632
  - Elements, Quick Start 44
  - Internal Responsibility To External Requirement 1167
  - Maintenance Item Between Categories 1716
  - Nudge Elements 88
  - Objects Between Packages 633
  - Operations Between Elements 647
  - Package Contents Up Or Down 455
  - Packages, Quick Start 44
  - Submenu 88
  - Test Between Categories 1703
- MS Word
  - Update RTF Report Links 1801
  - Use In RTF Documentation 1795
- MS Word Tables
  - Apply Styles 1799
  - Manipulate 1799
  - Resize 1799
- MSDE
  - Server Data Repository, Connect To 170
  - Server Repository, Create New 152
  - Upsize To 182
- MTS File
  - Advanced Options 1073
  - Create 1073
  - Incorporate Model Search 1073
  - Incorporate Model View 1073
  - Working With 1073
- Multi-page Diagram
  - Print 602
- Multiple Login
  - Under One User ID 203
- Multiple Select
  - Items From Project Browser 578
- Multiple Stereotype
  - Restrict Application Of 1054

- Multiplicity 522
    - Connector, Source Role 758
    - Connector, Target Role 760
    - Of Connector 776
    - Of Element 776
    - Of Element, Displayable 649
    - Of Element, Non-Displayable 663
  - MVC Pattern 1290
  - MyISAM
    - BaseModel Script 172
  - MySQL
    - Create Repository 149
    - Data Repository, Connect To 162
    - ODBC Driver, Set Up 154
    - Table Type, Set 1349
    - Upsize To 172
- N -**
- Name Template
    - Foreign Key 1365
    - Primary Key 1358
  - Namespace
    - And Whitespace Characters 1495
    - Clear 1495
    - Dialog 1495
    - Explanation 1495
    - List 1495
    - Locate In Project Browser 1495
    - Root 1495
    - Root Package Icon 454
    - Set 1495
  - Naming Format
    - Camel Case 1341
    - Pascal Case 1341
    - Spaced 1341
    - Underscored 1341
  - N-Ary
    - Association Element 1296
    - Element (Entity Relationship Diagram) 1382
  - Navigate
    - Diagram 473
  - Navigation And Selection
    - Hotkeys, For Diagram 582
  - Navigation Compass 103
  - Navigation Tools 440
  - Neaten Diagram Layout 614
  - Nested Version Control Packages 245, 248
  - Nesting
    - Connector 1002
    - Relationship 1002
  - New Action Dialog 866
  - New Code Sections
    - Add To Existing Features 1488
  - New Project
    - Menu Option (File Menu) 74
  - New Search Query 484
  - New Structured Activity Dialog 926
  - Node
    - Element 956
  - Non-Selectable
    - Element 655
  - Normal.rtf Template File 1734
  - Notation
    - Co-Region 996
    - Process Modeling 1195
  - Note
    - Add To Link/Connector 743
    - Attach To Connector 624
    - Create Project Glossary Item 769
    - Element 913
    - Element, Create 639
    - For Attribute 769
    - For Connector 769
    - For Diagram 769
    - For Element 769
    - For Operation 769
    - Formatting 770
    - Hyperlink Glossary Term 769
    - Insert Glossary Item 769
    - Insert New From Toolbar 110
    - Keyboard Shortcuts 770
    - Link To Element Feature 640
    - Link To Internal Documentation 640
    - Spell Check 769
    - Tab 769
    - Toolbar Options 770
    - Window 769
  - Notelink
    - Connector 1003
    - Insert New From Toolbar 110
    - Relationship 1003
  - Notes
    - Gap, Edit 512
  - Nudge
    - Elements 88, 600
  - Numbered Sections
    - In RTF Reports 1761
  - Numbering Levels
    - Apply 1761
    - In Virtual Documents 1782
    - RTF Reports 1761
    - Use 1761

- Numbering List
    - RTF Reports 1761
  - Numeric Range Generator
    - Timeline Element States 840
  - NUnit Transformation
    - MDA-Style Transform 1323
- O -**
- Object
    - Appearance, Options 432
    - Attributes 1333
    - Classes 1333
    - Classifiers 705
    - Columns 1333
    - Connect 741
    - Connector 666
    - Create From Attribute 704
    - Definition 1333
    - Element 957
    - Elements And Connectors 555
    - Group, Toolbox 555
    - Instance 957
    - Links 666
    - Move Between Packages 633
    - Multiplicity 758
    - Operations 1333
    - Packages 1333
    - Parameters 1333
    - Properties 660, 1333
    - Relationships 666
    - Scenario 667
    - State, Set 959
    - Tables 1333
    - Transformation 1333
    - Type 1333
    - Type, Change For Element 635
  - Object Constraint Language
    - Model Validation 1688
    - Model Validation Rules For Conformance 1692
  - Object Diagram
    - Description 799
    - Elements And Connectors 799
    - Example 800
  - Object Flow
    - Connector 1003
    - In Activity Diagram 1004
    - In State Machine Diagram 1003
    - Multiple 1004
    - Relationship 1003
    - Selection Behavior 1004
    - Simple 1004
    - Transformation Behavior 1004
    - With Action Pins 1004
  - Object Oriented Programming
    - C Code Generation For UML Model 1462
    - Limitations 1462
  - Object Workbench 1663
    - Constraints 1665
    - Introduction, Visual Execution Analyzer 1663
    - Invoke Method 1666
    - Modes 1664
    - Overview, Visual Execution Analyzer 1664
    - Platforms Upported 1664
    - Requirements 1665
    - Workbench Variables, Constructors 1665
    - Workbench Variables, Create 1665
    - Workbench Variables, Delete 1665
  - ObjectType Enum
    - Automation Interface 1835
  - Occurrence
    - Connector 1005
    - Element 906
    - Relationship 1005
  - OCL
    - Model Validation 1688
  - OCL Constraints
    - Attribute 1692
    - Element 1692
    - Feature 1692
    - Model Validation Rules for Conformance 1692
    - Relationship 1692
  - ODBC
    - Data Modeling 1379
    - Driver, Set Up 154
  - ODBC Data Source
    - Select 1377
  - ODBC Source
    - Select Database Objects From 1377
    - Select Stored Procedures From 1377
    - Select Tables From 1377
  - Offline
    - Checkout 250
    - Version Control 250
  - OLE DB Provider
    - Microsoft For Oracle 166
    - Oracle Provider 166
  - on the File | Document Options menu 1767
  - Online Resources
    - Access From Learning Center 72
    - Submenu 101
  - Open
    - External Tools 121

- Open
  - Package From Project Browser 533
  - Package Within Diagram 585
  - Report in Microsoft Word 1796
- Open Office
  - Optimize For, In RTF Report Generation 1732
- Open Project
  - Menu Option (File Menu) 74
- Open Repository
  - Automation Interface Code Example 1953
- Open Source Directory
  - Element Code 88
- Open Source File
  - Menu Option 90
- Operation 454
  - Add To Element, In-place Editor 726
  - Appearance 707
  - As Action 865, 872
  - Associate With Behavior 710
  - Behavior Description, Show In Diagram 710
  - Behavior, Initial Code 711
  - Business Rule 1209
  - Connect To 742
  - Constraints 711
  - Context Menu, Project Browser 452
  - Copy Between Elements 646
  - Create Use Case For 1406
  - Definition 707
  - Dialog, Behavior Tab 710
  - Dialog, General Tab 708
  - Dialog, Post Tab 711
  - Dialog, Pre Tab 711
  - Disconnect From 742
  - Display Inherited Operation 713
  - Edit Name, In-Place Editor 722
  - Edit Parameter Kind 725
  - Edit Scope 723
  - Edit Stereotype, In-place Editor 722
  - Element Feature 707
  - Fast Create 708
  - Implement Interface Operations 712
  - Implement Parent Operations 712
  - In Project Browser 707
  - Inherited, Show 585
  - Introduction 707
  - Link To Use Case 1406
  - Move Between Elements 647
  - Of State Element, Behavior 710
  - Override Interface Operations 712
  - Override Parent Operations 712
  - Parameter Keyword, Edit 724
  - Parameter, By Reference 719
  - Parameter, Insert In Element 726
  - Parameters, Define 717
  - Private, Icon 454
  - Properties, Behavior 710
  - Properties, General 708
  - Properties, Initial Code 711
  - Properties, Postconditions 711
  - Properties, Preconditions 711
  - Protected, Icon 454
  - RuleTask, Behavior Call Action 1209
  - Show On Diagram 585
  - Tagged Values, Add 712
  - WSDL Port Type Operation 1615
- Operation Constraints Dialog 1679
- Operational Domain Model
  - SysML, Create 1573
- Options
  - Audit 300
  - Compare Utility 314
  - Element Visibility 434
  - Read-Only, For Enterprise Architect 18
  - Reset For A Class 1540
  - Visual Styles/Themes 125
- Options Dialog
  - ActionScript 1530
  - Ada 2005 1531
  - Appearance Options 427
  - Attribute/Operation Specifications 1522
  - C 1531
  - C# 1532
  - C++ 1533
  - Communication Message Coloring 435
  - Connector Settings 434
  - Delphi 1533
  - Diagram Behavior 430
  - Diagram Settings 426
  - General Settings 422
  - Introduction 421
  - Java 1535
  - Links Settings 434
  - MDG Technology 1540
  - Object Appearance 432
  - Object Appearance, Default Fonts 429
  - Object Appearance, Element Visibility 434
  - PHP 1536
  - Python 1536
  - Sequence Diagram Options 431
  - Standard Colors 424
  - Standard Colors - Set Project Colors 425
  - SystemC 1537
  - VB.NET 1537
  - Verilog 1538

- Options Dialog
  - VHDL 1538
  - Visual Basic 1539
  - XML Specifications 436
- Oracle
  - ODBC Driver, Set Up 160
  - Package, Create 1373
  - Sequence 1351
  - Sequence Options, DDL For Packages 1380
  - Sequence Options, DDL For Table 1379
  - Tables, Set Properties 1350
  - Tables, Tagged Values 1350
  - Temporary Table 1350
  - Upsize To 177
- Oracle 10g
  - Data Repository, Connect To, Via ODBC 164
  - Data Repository, Connect To, Via OLE DB 166
  - Server Repository, Create 151
- Oracle 11g
  - Data Repository, Connect To, Via ODBC 164
  - Data Repository, Connect To, Via OLE DB 166
  - Server Repository, Create 151
- Oracle 9i
  - Data Repository, Connect To, Via ODBC 164
  - Data Repository, Connect To, Via OLE DB 166
  - Server Repository, Create 151
- Order
  - Enterprise Architect 26
  - Package Contents 455
- Outline
  - Red 214
- Output
  - Business Modeling 1197
  - Debugger, View 1446
  - Debugger, Window 1446
- Output Window
  - Audit History Tab 307
  - Context Menu 127
- Override
  - Attribute Initializer 702
  - Default Toolbox In Toolbox Profile 1076
  - Implement Parent Operations 712
  - Parent Operations 712
- Overrides & Implementations
  - Of Classes And Interfaces 86
- Overview
  - Visual Execution Analyzer 1635
- Add And Manage, Automation Interface Code Example 1954
- Add Element Directly 627
- Add To Diagram From Toolbox 534
- Add To Project Browser 534
- Add To UML Model, Quick Start 39
- Apply Version Control To Model Branch 289
- As RTF Bookmark 1796
- Automation Interface 1829
- Automation Interface Repository 1845
- Baselines 308
- Batch Export To XMI 332
- Batch Import From XMI 333
- Body, For Oracle 1373
- Comparison, Example 315
- Configuration 329
- Configure For Version Control 282
- Context Menu, Project Browser 444
- Control, Menu 328
- Control, Remove 331
- Controlled 282
- Controlled, Report Deletion Of Cross Package References in XMI 335
- Copy 535
- Copy Between Projects 344
- Create Diagram 568
- Create Oracle Packages 1373
- CSV Import/Export Specification 338
- Delete In Project Browser 537
- Duplicate 535
- Element 960
- Existing, Drag Onto Diagram 536
- Export 327
- Export To XMI Stubs 329
- Export Version Controlled Model Branch 290
- Hide Contents On Diagram 537
- Icon Overlays 454
- Import 327
- Import Version Controlled Model Branch 291
- Load 332
- Lock 212
- Lock, Require User Lock 213
- Lock, User/Group Lock 210
- META-INF 1314
- Modeling With 533
- Move 44
- Move Element Between 633
- Move Objects Between 633
- Nested in Version Control 248
- Open From Diagram 585
- Open From Project Browser 533
- Paste Copy 535

**- P -**

Package

- Package
  - Phase, Update 411
  - Profile 1041
  - Rename 534
  - Resynchronize Package Version Control Status 297
  - Review Version Control History (SCC) 294
  - Review Version Control History (TFS, CVS, Subversion) 293
  - Save 331
  - Save Profile 1050
  - Show Contents On Diagram 537
  - Specification, For Oracle 1373
  - Status, Update 411
  - Synchronize Contents 1494
  - Tasks 533
  - Update Contents 1494
  - Validate Version Control Configuration 296
  - Version Control 242
  - Version Control Revision, Check Out 293
  - Version, Update 411
  - Working With 533
  - XML 327
- Package Browser
  - Context Menu Options 459
  - Description 456
  - Filter Bar 458
  - Generate RTF Report 1727
  - Insert Tagged Value Column 459
  - Options 459
  - Toolbar 459
  - Value Grouping 458
  - View Header 458
  - Work On Elements 459
- Package Context Menu
  - Add Submenu, Project Browser 446
  - Build And Run Submenu, Project Browser 448
  - Code Engineering Submenu, Project Browser 447
  - Contents Submenu, Project Browser 448
  - Documentation Submenu, Project Browser 447
  - Import/Export Submenu, Project Browser 448
- Package Control
  - Options, Dialog 329
- Package Diagram
  - Description 796
  - Elements And Connectors 796
  - Example 797
- Package Import
  - Connector 1006
  - Relationship 1006
- Package Merge
  - Connector 1006
  - Relationship 1006
- Package Scenarios
  - Import As Test Scenarios 1703
- Package Structure
  - BPEL 1238
- Packaging Component
  - Element 1297
- Page
  - Footer, Print On Diagram 571
  - Header, Print On Diagram 571
  - Setup, Menu Option (File Menu) 74
  - Size, Set For Diagram 600
- Pan
  - Diagram View 599
- Pan And Zoom
  - Diagram 473
  - Window 473
- Paragraph Numbering
  - In RTF Style Template Editor 1751
- Parameter 719
  - Activity 717
  - Automation Interface, ElementFeatures Package 1906
  - Behavior 717
  - Behavior Call Arguments 716
  - Behavior, Tagged Values 718
  - Behavioral, Edit 717
  - Behavioral, Extend 717
  - Behavioral, Reassign 717
  - Behavioral, Set 717
  - Behavioral, Synchronize Call Argument With 716
  - Behavioral, Synchronize Invocation Argument With 716
  - Dialog 717
  - For Rule Flow Activity 1208
  - Interaction 717
  - Kind Inout, By Reference 719
  - Operation 717
  - Operation, By Reference 719
  - Reference 719
  - Show Details On Diagram 575
  - Substitution 1011
  - Variables In Business Rules 1208
- Parameter Information
  - Display In Source Code 1410
- Parameter Kind
  - Edit for Element Operation 725
- Parameter Substitution
  - Binding Connector 1011
- Parameterized Class

- Parameterized Class
  - And Binding Connector 1010
- Parameterized Classes (Templates) 943
- Parametric
  - Diagram 1568
  - Diagram, Simulate SysML Model 1570
  - Model, SysML, Create 1568
- ParamTag
  - Automation Interface, ElementFeatures Package 1908
- Parent
  - Confirm Element As Parent 632
  - Set For Element 628
- Parent And Interfaces
  - Elements 86
- Parse
  - Source Code Files In Viewer 1408
- Part
  - Add Property Value 961
  - Element 960
  - Property Tab 963
  - Qualifiers 963, 972
  - Redefined 963
  - Represent On Sequence Diagram 858
  - Subsetted 963
- Partial Class
  - Generate 1493
- Partition
  - Activity 914
  - Docking 914
  - Element 914
  - Horizontal 914
  - Vertical 914
- Partitions Collection
  - Automation Interface, ElementFeatures Package 1908
- Pascal Case
  - Naming Format 1341
- Password
  - Administrator Change 209
  - Administrator Set 209
  - Security, Change 198
  - User Change 209
- Password Encryption
  - Prior To Release 7.1 Of Enterprise Architect 208
- Password Encryption (Repository)
  - At Release 7.1 Of Enterprise Architect 140, 143
- Paste
  - Activity As Action 579
  - Activity As Link 579
  - Composite Element 578
  - Connectors Between Pasted Activities 579
  - Connectors Between Pasted Instances 576
  - Copy Of Element 634
  - Copy Of Package 535
  - Diagram Into Package 583
  - Element From Project Browser 576
  - Element On Diagram 579
  - Multiple Items As Children 578
  - Multiple Items As Instances 578
  - Multiple Items As Links 578
  - Multiple Items From Project Browser 578
  - Object As Child 576
  - Object As Link 576
  - Object As New Instance 576
- Paste Element Dialog 576
- Paste Elements
  - As Link 75
  - As New 75
  - From Clipboard As Metafile 75
  - Menu Option (Edit Menu) 74
  - Submenu (Edit Menu) 75
- Patches
  - SQL, Run 418
- Pattern
  - Action, Modify 1024
  - Actions 1022
  - Add To MDG Technology 1067
  - Create From Diagram 1022
  - Default, Change 1024
  - Design 1021
  - Gang of Four 1021
  - GoF 1021
  - GoF, Download 1023
  - Import Into Model 1023
  - In Resources View 1022
  - Model-View-Controller 1290
  - MVC 1290
  - Save 1022
  - Save From Diagram 1022
- PDATA
  - Diagram Profile Attribute Values 1081
  - Element Attribute In MiscData, Object Model 1877
- People
  - As Project Resources 781
  - Assign To Changes 1722
  - Assign To Defects 1722
  - Dialog 778
  - Project Types Menu Option 99
  - Settings 778
- Per Page Diagram Layout 611

- Performance
  - Of Enterprise Architect 162, 163, 164, 166, 167, 169, 170, 183, 298
- Permission List
  - User Security 205
- Personal Information
  - Allocated Work 377
  - Mail, Sent & Received 377
  - Project Tasks 377
  - Window 377
  - Workflow 377
  - Working Sets 377
- Personal Information Window
  - Allocated Work Tab 382
  - Execute Workflow Scripts 386
  - Monitor Tasks 385
  - Project Tasks Tab 385
  - Record Work Progress 382
  - Resource Allocation 382
  - Workflow Tab 386
  - Working Sets Tab 387
- PHP
  - Code Generation 1536
  - Import, Reverse Engineering 1510
  - Language Options 1536
  - Modeling Conventions 1472
- PHP Debugger
  - Checklist 1434
  - Local 1432
  - Remote 1432
  - Scenarios 1432
  - Script Set Up 1432
  - Supported Operating Systems 1433
  - System Requirements 1433
  - Timeouts 1434
- Physical Data Model 1346
- Physical Model 526
- PI Macro
  - Code Template Syntax 1133
- PIM
  - Internal Bindings 1301
- Pin
  - Action 869
  - Add To Action 869
  - As Action Property 870
  - As Argument For Call Action 869
  - Assign To Action 869, 870
  - Connector End Point 114
  - Connector Start Point 114
  - Properties 869
  - Toolbox Pages 549
- Pkg Import
  - Connection 1006
  - Relationship 1006
- Pkg Merge
  - Connector 1006
  - Relationship 1006
- Place Recording Markers
  - Execution Analysis, Recording Sequence Diagrams 1642
- Platform Naming Conventions 1341
- Platform Specific Model 1301
- Platform-Independent Model 1301
- Pool (BPEL)
  - Create 1249
  - Model 1249
- Port
  - Add To Element 962
  - Element 962
  - For Trigger Element 934
  - Inherited By Component Instance, From Classifying Component 963
  - Inherited By Component, From Realized Interface 963
  - Property Tab 963
  - Qualifiers 963, 972
  - Redefined 963
  - Represent On Sequence Diagram 858
  - Subsetted 963
- Port Type
  - WSDL Diagram 1614
  - WSDL Element 1614
- Port Type Operation
  - WSDL 1615
- Position
  - Options For Elements In Diagram 85
- Position Elements 88
- Post
  - Add To Team Review 222
  - Create In Team Review 222
  - Create Reply 224
  - Delete 218
  - Delete In Team Review 226
  - Edit In Team Review 226
  - Reply To In Team Review 224
- Post-Constraint
  - RTF Report Template Section 1741
- PostgreSQL
  - Data Repository, Connect To 167
  - ODBC Driver, Set Up 155
  - Repository, Create 151
  - Upsize To 179
- Post-New Events
  - Add-In Model 2010

- Post-New Events
  - EA\_OnPostNewAttribute 2014
  - EA\_OnPostNewConnector 2011
  - EA\_OnPostNewDiagram 2012
  - EA\_OnPostNewDiagramObject 2013
  - EA\_OnPostNewElement 2011
  - EA\_OnPostNewGlossaryTerm 2016
  - EA\_OnPostNewMethod 2015
  - EA\_OnPostNewPackage 2015
- Pre/Post Conditions
  - Constraints 871
  - Local 871
  - Notes 871
  - On Actions 871
- Precede
  - Connector 553
- Pre-Constraint
  - RTF Report Template Section 1741
- Predefined Tag Type
  - Assign To Stereotype 1045
  - Define 1045
- Predefined Tagged Value Type
  - Filters 1108
  - Reference Data 1111
  - Structured 1108
  - Syntax 1108, 1111
- Pre-Deletion Events
  - Add-In Model 2017
  - EA\_OnPreDeleteAttribute 2018
  - EA\_OnPreDeleteConnector 2020
  - EA\_OnPreDeleteDiagram 2020
  - EA\_OnPreDeleteDiagramObject 2021
  - EA\_OnPreDeleteElement 2017
  - EA\_OnPreDeleteGlossaryTerm 2023
  - EA\_OnPreDeleteMethod 2019
  - EA\_OnPreDeletePackage 2022
- Pre-New Events
  - Add-In Model 2023
  - EA\_OnPreNewAttribute 2027
  - EA\_OnPreNewConnector 2025
  - EA\_OnPreNewDiagram 2026
  - EA\_OnPreNewDiagramObject 2026
  - EA\_OnPreNewElement 2024
  - EA\_OnPreNewMethod 2028
  - EA\_OnPreNewPackage 2029, 2030
- Preprocessor Macros 1525
- Preserve Hierarchy
  - CSV Specification 340
- Primary Key
  - Complex 1358
  - Create 1358
  - Description 1358
  - Extended Properties 1360
  - Name Template, Define 1358
  - Simple 1358
  - SQL Server, Non-Clustered 1360
- Primitive
  - Element 964
- Print
  - Diagram (Single) From Project Browser 451
  - Diagrams (Multiple) From Project Browser 451
  - Menu Option (File Menu) 74
  - Multi-page Diagrams 602
  - Page Footer On Diagram 571
  - Page Header On Diagram 571
  - Preview, Menu Option (File Menu) 74
  - Project Issues 358
  - Scaled Image 602
  - Setup, Menu Option (File Menu) 74
  - Task List 356
- Print Preview
  - Display 540
  - Multiple Pages 540
- Private Key
  - Add 2096
- Problem Type
  - Define 787
- Process
  - BPEL, Properties 1240
  - Element 1195, 1297
  - Element (Data Flow Diagram) 1184
  - Model Template 520
  - Modeling 520
- Process Memory
  - Inspect 1445
- Process Modeling Notation 1195
- Professional Edition
  - Of Enterprise Architect 14
  - Upsize From 148
- Profile
  - Add Attribute To Diagram 1028
  - Add Connector To Diagram 1028
  - Add Element To Diagram 1028
  - Add Elements 1042
  - Add Enumeration Elements 1047
  - Add Metaclasses 1042
  - Add Operation To Diagram 1028
  - Add Shape Script 1048
  - Add Stereotypes 1042
  - Add To MDG Technology 1066
  - And Element Templates 643, 1026, 1039
  - Application, Analyzer menu Option 94
  - Constraints 1046
  - Create 1041

- Profile
  - Diagram, Create 1080
  - Elements And Connectors 560
  - Export To Disk 1050
  - Group, Toolbox 560
  - Import 1027
  - Import From XML 1026, 1039
  - Learning Center, Create 1082
  - Package 1041
  - Save 1050
  - Save From Diagram 1050
  - Save From Package 1050
  - Set Default Appearance Of Stereotype Objects 1049
  - Stereotype 1041, 1058
  - Stereotypes 1026, 1039
  - Swimlanes Matrix 592
  - Tagged Values 1028
  - Tags 1044
  - Toolbox 1074
  - Work With 1041
  - Zachman 592
- Profile Connector
  - Add To Diagram 1028
- Profile Diagram
  - Description 809
  - Example 810
- Profile Element
  - Add To Diagram 1028
- Profiler
  - Attach To Process 1660
  - Getting Started 1659
  - Launch 1660
  - Operation 1661
  - Overview 1658
  - Prerequisites 1658
  - Report, Example 1658
  - Report, Load 1662
  - Report, Save 1662
  - Report, Save As Resource In Team Review 1662
  - Set Options 1661
  - Set Sample Intervals 1661
  - Start 1660
  - Stop 1660
  - Supported Platforms 1658
  - System Requirements 1658
  - Team Review, Save Report As Resource 1662
  - Toolbar 1659
- Progress OpenEdge
  - Data Repository, Connect To 170
  - ODBC Driver, Set Up 159
- Server Repository, Create 153
- Upsize To 182
- Project
  - Administration, Security Permissions 196
  - Author 778
  - Browser 441
  - Change Control 1710
  - Clean 414
  - Clean Prior To Build 1416
  - Clients 782
  - Compact .EAP File 418
  - Comparison With Other Project 337
  - Configure 145
  - Constants 1772
  - Copy Existing 147
  - Create In Enterprise Architect, Tutorial 37
  - Create, Quickstart Tutorial 37
  - Data Integrity 414
  - Data, Transfer 343
  - Develop In Team Environment, Introduction 185
  - Development, Introduction 135
  - EABase 35
  - Estimation 400
  - Explorer 441
  - File, EABase 145
  - Glossary 356, 362
  - Integrity Check 414
  - Integrity, Run SQL Patches 418
  - Issues 356
  - Issues, Record 358
  - Items, Add Via Toolbar 108
  - Maintenance 1710
  - Menu 80
  - Metrics 400
  - Model Template 528
  - Open Existing 136
  - Recover 414
  - Remove Recent 71
  - Rename .EAP File 418
  - Repair Project .EAP File 419
  - Resources 781
  - Roles 780
  - Share, DBMS Repository 186
  - Share, Network Drive 186, 187
  - Share, Replication 186
  - Sharing, Introduction 185
  - Spell Checking 372
  - Statistics, View 80
  - Structure 515
  - Tasks 57, 356
  - Timescale Estimation 400
  - Toolbar 108

- Project
  - Transfer 172
  - Upgrade 417
  - View 441
  - What Is A? 137
- Project Browser 448
  - Attribute Context Menu 452
  - Collapse Contents 448
  - Context Menus 443
  - Default Behaviour 455
  - Diagram Context Menu 451
  - Element Context Menu 448
  - Element Context Menu, Add Submenu 450
  - Exclamation Marks 214
  - Expand Contents 448
  - Free Sorting 455
  - Hide And Show 441
  - Icon Overlays 454
  - Icon, User-Defined 1051
  - Introduction 441
  - Method Context Menu 452
  - Model Context Menu 443
  - Move Items Within 441
  - Open Package 533
  - Operation Context Menu 452
  - Order Package Contents 441
  - Package Context Menu 444
  - Package Context Menu, Add Submenu 446
  - Package Context, Build And Run Submenu 448
  - Package Context, Code Engineeing Submenu 447
  - Package Context, Contents Submenu 448
  - Package Context, Documentation Submenu 447
  - Package Context, Import/Export Submenu 448
  - Reload Current Package 448
  - Reset Sort Order 448
  - Scripts 1819
  - Selective Collapse of Packages 441
  - Show Level Numbering 444
  - Show Stereotypes 455
  - Toolbar 452
  - Version Control Indicators 249
  - Views 441
- Project Calendar
  - Allocated Resources View 396
  - Calendar Panel 390
  - Calendar Panel, Configure Event Subtypes 395
  - Diary Panel 390
  - Events 392
  - Options 390
  - Overview 390
  - Project Tasks View 398
  - Recurring Events 392
  - Toolbar 390
  - View, Calendar 392
- Project Constant
  - RTF Reports 1772
- Project Custom Colors
  - Get For Element 642
  - Set For Element 642
- Project Discussion Forum
  - Now Team Review 216
- Project Factor Calibration 400
- Project File
  - Create 136
  - Example 138
  - Open 138
- Project Glossary
  - Add Item, Glossary Detail Dialog 362
  - Add Item, Glossary Dialog 363
  - Create Item From Notes Text 769
  - Delete Item, Glossary Dialog 363
  - Delete Item, Project Glossary Tab 362
  - Filter List, Glossary Dialog 363
  - Filter List, Project Glossary Tab 362
  - Glossary Dialog 363
  - Glossary Report 364
  - Hyperlink Term From Notes 769
  - Insert Item In Text 769
  - Modify Item, Glossary Detail Dialog 362
  - Modify Item, Glossary Dialog 363
  - Redefine Entry Type, Project Glossary Tab 362
  - Tab 362
- Project Indicators
  - Project Types Menu Option 99
  - Risk Types 355
- Project Interface
  - Automation Interface 1929
  - Project 1929
- Project Issue
  - Add 358
  - Delete 358
  - Dialog 358
  - Modify 358
  - Print 358
  - Record 358
  - Report, Via Project Issues Dialog 360
  - Report, Via Project Issues Tab 360
  - Tab 358
- Project Management
  - Asterisk On Window Tabs 348
  - Default Hours 404

- Project Management
  - Effort Management 350
  - Effort Types 353
  - Environment Complexity Factors 402
  - Introduction 346
  - Maintenance 1713
  - Metric Types 354
  - Metrics 347, 352
  - Resource Allocation 349
  - Resource Report 370
  - Resources 347
  - Risk Management 351
  - Risk Types 355
  - Risks 347
  - Technical Complexity Factors 401
  - Toolbar 347
  - Window 347, 348
  - With Enterprise Architect 346
- Project Manager
  - And Enterprise Architect 63
  - Project Estimation 63
  - Project Role 63
  - Resource Management 63
  - Risk Management 63
- Project Role
  - And Enterprise Architect 56
  - Business Analyst 58
  - Database Administrator 67
  - Deployment and Rollout 65
  - Developer 61
  - Implementation Manager 65
  - Project Manager 63
  - Software Architect 59
  - Software Engineer 60
  - Technology Developer 65
  - Tester 64
- Project Settings
  - Configure (Settings Menu) 98
- Project Task
  - Add 357
  - Delete 357
  - Description 356
  - List 356
  - Modify 357
  - Tab, Print List 356
- Project Task Allocation
  - Element View 368
  - Resource View 366
  - Resource Report 370
  - Window 366
- Project Tasks
  - Create, Project Calendar 398
  - Create, Personal Information Window 385
  - Delete, Project Calendar 398
  - Delete, Personal Information Window 385
  - Edit, Project Calendar 398
  - Edit, Personal Information Window 385
  - Monitor, Project Calendar 398
  - Monitor, Personal Information Window 385
  - Tab, Personal Information Window 385
  - View, Project Calendar 398
- Project Team Review
  - Access 216
  - Add External File Link To Post 227
  - Add New Category 220
  - Add New Post 222
  - Add Object Link To Post 227
  - Add Team Review Link To Post 227
  - Connections To Other Team Reviews 230
  - Copy Path To Clipboard 218
  - Create Category 220
  - Create Post 222
  - Create Reply 224
  - Delete Item 226
  - Edit Item 226
  - Editor 226
  - Icons 216
  - Introduction 216
  - Load Data When Required 229
  - Loading Behavior 229
  - Mark All Posts Unread 229
  - Options 229
  - Preload 229
  - Reply To Post 224
  - Search 229
  - Was Discussion Forum 216
- Project Types
  - Estimation Factors 99
  - General Types 99
  - Maintenance Data 99
  - People 99
  - Project Indicators 99
  - Submenu 99
- Project Version Control Menu 277
- ProjectIssues
  - Automation Interface Repository 1851
- ProjectResource
  - Automation Interface Repository 1852
- ProjectRole
  - Automation Interface Repository 1853
- Properties
  - Automation Interface, ElementFeatures Package 1909
  - Behavior Tab 715

- Properties
    - Call Tab 715
    - Connector, Advanced 761
    - Connector, Menu Section 737
    - Constraints, Scenario 686
    - Dialog, Element 660
    - Editor Language 1520
    - Effect Tab 865
    - Element 660
    - Element Context Menu 649
    - Element, Advanced 664
    - Element, Associated Files 687
    - Element, Connectors 666
    - Element, Constraints 666
    - Element, Details 663
    - Element, External Requirements 665
    - Element, General 662
    - Element, Internal Requirements 664
    - Element, Links 666
    - Element, Relationships 666
    - Element, Requirements 664
    - Element, Trigger Tab 934
    - Extend For Requirements 1160
    - Object 660
    - Of Classifiers, Composite Structure Diagram 803
    - Of Requirements, Extended 1160
    - Of Requirements, Standard 1157
    - Requirement, Display On Diagram 1160, 1161
    - Window (Element) 689
  - Properties Note
    - Diagram 587
  - Properties,
    - Part, Property Tab 963
    - Port, Property Tab 963
  - Properties, EASL
    - Action 1141
    - Argument 1141
    - Behavior 1141
    - Call Event 1141
    - ChangeEvent 1141
    - Classifier 1141
    - Condition 1141
    - Construct 1141
    - Edge 1141
    - EventObject 1141
    - Instance 1141
    - Parameter 1141
    - Primitive 1141
    - PropertyObject 1141
    - SignalEvent 1141
    - State 1141
    - StateMachine 1141
    - TimeEvent 1141
    - Transition 1141
    - Trigger 1141
    - Vertex 1141
  - Property
    - Automation Interface, ElementFeatures Package 1909
    - Redefined 692
    - Select, Dialog 692
    - Subsetted 692
  - Property Validation
    - Attribute 1692
    - Element 1692
    - Feature 1692
    - Relationship 1692
  - Property Value
    - Part, Add To 961
  - PropertyType
    - Automation Interface Repository 1853
  - PropType Enum
    - Automation Interface 1836
  - Proxy
    - (Shortcut) File 140
  - Pseudo-State
    - Elements 820
    - Fork 898, 900
    - In State Machine Diagram 820
    - Join 898, 901
  - PSM 1301
  - Purchase
    - Enterprise Architect 26
  - Python
    - Code Generation 1536
    - Import, Reverse Engineering 1510
    - Language Options 1536
    - MDG Technology For, Enterprise Architect 1036
    - Modeling Conventions 1473
- Q -**
- Qualified Association 971
  - Qualifier
    - Association End 972
    - Association Property 971
    - Attribute 972
    - Attribute Property 971
    - Dialog 972
    - Of Attribute 698
    - Of Port 963
    - Part 972

- Qualifier
    - Part Property 971
    - Port 972
    - Port Property 971
    - Set Properties 972
  - Quality Control 1687
    - Model Validation 1688
    - Spell Checking 372
    - Testing 1694
  - Query Builder
    - Search Definition 484
  - Query Methods
    - In Shape Scripts 1099
  - Quick Add
    - Tagged Values 763
  - Quick Linker 741
    - Arrow 622
    - Connector Names, In Definitions 1062
    - Create Connector 624
    - Create Element And Connector 623
    - Default Settings, Hide 1062
    - Definition Format 1059
    - Element Names, In Definitions 1062
    - Example 1061
    - Introduction 622, 1058
    - Options 623
  - Quick Start
    - Add Connectors To UML Model 42
    - Add Diagram To Package 40
    - Add Diagram To UML Model 40
    - Add Element To Diagram 41
    - Add Element To UML Model 41
    - Add Package to UML Model 39
    - Auditing 300
    - Create A Project In Enterprise Architect 37
    - Define Connector Properties 43
    - Define Element Properties 43
    - Delete UML Model Components 48
    - Generate HTML Report 1802
    - Generate RTF Report 1727
    - Move Project Components 44
    - Project Tasks 57
    - Save Project Changes 52
    - View HTML Report 1802
  - Quickstart
    - Common Tasks 36
- R -
- Rational Rose
    - And XML 319
    - Export To 325
  - Rational Software Architect
    - Models 324
  - Rational Software Modeler
    - Import \*.emx Files 323
    - Import \*.uml2 Files 323
  - Read-Only Options
    - For Enterprise Architect 18
  - Realization
    - Connector, Quick Generation Of 1161
  - Realization Link
    - Implement Parent Operations 712
    - Override Parent Operations 712
  - Realize
    - An Information Flow 985
    - Connector 1007
    - Relationship 1007
  - Realized Interfaces
    - For Class, Show On Diagram 597
  - Receive
    - Element 915
    - Event 1292
  - Recent Discussions
    - Concerning Model View Items 464
  - Recent Post Options Dialog 467
  - Recent Postings
    - Display 469
  - Reception
    - Definition 964
    - Of Signal 964
  - Record
    - Application, Analyzer menu Option 94
    - Submenu (Analyzer) 96
  - Record & Analyze Window 1651
  - Record Activity For Class
    - Execution Analysis, Record Sequence Diagram 1646
  - Record Activity For Method
    - Execution Analysis, Record Sequence Diagram 1645
  - Record Macro
    - In Source Code Editor 1410
  - Record Sequence Diagrams
    - Control Recording 1648
  - Record State Changes
    - Visual Execution Analyzer 1657
  - Record Work Progress
    - Personal Information Window 382
  - recording 1650
  - Recording Actions
    - Create Sequence Diagram, Call Stack 1450
    - Debugger, Overview 1449

- Recording Actions
  - Debugger, Step Through Function Calls 1449
  - Save Call Stack 1450
- Recording History
  - Save, Execution Analysis 1653
- Recording Markers
  - Activate, Execution Analysis 1647
  - Breakpoints And Markers Window, Execution Analysis 1646
  - Difference From Breakpoint 1642
  - Disable, Execution Analysis 1647
  - Marker Types, Execution Analysis 1642
  - Place, Execution Analysis 1642
  - Work With Marker Sets, Execution Analysis 1648
- Recover
  - Controlled Package 334
  - Project 414
- Rectangle Notation
  - Element Menu Option 649
  - For Shaped Elements 938
- Recursion
  - Connector 1008
  - Message 1008
  - Relationship 1008
- Recursive Builds
  - Visual Execution Analyzer 1416
- Red
  - Border 214
  - Exclamation Mark 214
  - Letter A 729, 730
  - Object Outline 214, 602, 655
  - Triangle 412
- Redefined
  - Property, Port 963
- Redefined Port
  - Manage 963
- Redefined Property
  - Of Attribute 698
- Redefinition
  - Connector 560
- Redo
  - Last Action, Diagram Edits 603
  - Option (Edit Menu) 74
- Re-entrancy
  - In Add-Ins 1972
- Reference
  - Automation Interface 1828
  - Automation Interface Repository 1854
- Reference Data 772
  - And Version Control 246
  - Cardinality (Multiplicity) 776
  - Clients 782
  - Constraint Status Types 785
  - Constraint Types 784
  - Estimation 787
  - Export 237
  - Export, Introduction 236
  - General Types 782
  - Import Automatically 239
  - Import Manually 239
  - Import, Introduction 236
  - Maintenance 787
  - Metrics 787
  - People 778
  - Problem Types 787
  - Project Author 778
  - Requirement Types 785
  - Resources 781
  - Roles 780
  - Scenario Types 786
  - Share 236
  - Status Types 783
  - Stereotypes 773
  - Tagged Value Types 775
  - Testing Types 788
  - UML Types 772
- Reference Data Tagged Value Type 1111
- Reference Data Tagged Values
  - Create 1113
- Referenced XML Schema
  - Import 1600
- Refresh
  - Diagram 187
  - Image 593
  - Project 187
  - View Of Shared Model 187
- Region
  - Composite State 820
  - Concurrent Substate 820
  - Element 916
  - Expansion, Element 893
  - Interruptible Activity, Element 907
  - On Composite State 919
  - State Machine 820
- Register
  - Add-Ins 2101
  - Enterprise Architect 29
- Registration Key
  - In License Information 2095
- Registry Settings 118
  - User 421
- Regular Expressions
  - Code Editor Find And Replace 1412

- Regular Expressions
  - Metasequences 1412
  - Tagged Regions 1412
- Related Elements
  - Find 580
  - Insert With Context Menu 652
  - Place On Current Diagram 580
- Relational Database
  - Model Template 524
- Relationship 1382
  - Activity Edge 985
  - Aggregate 968
  - Application 560
  - ArchiMate 1272
  - Assembly 969
  - Associate 970
  - Association 970
  - Association Class 973
  - Asynchronous Signal 996
  - Communication 977
  - Communication Path 975
  - Compose 976
  - Composite Aggregation 976
  - Connector 977
  - Control Flow 977
  - Create Using Relationship Matrix 500
  - Data Flow 1184
  - Delegate 978
  - Delete 666
  - Delete Using Relationship Matrix 500
  - Dependency 979
  - Dependency, Apply Stereotype 979
  - Deployment 980
  - Display Options 434
  - Element (Entity Relationship Diagram) 1382
  - Entity Relationship Diagram 1382
  - Eriksson-Penker 1273
  - Extend 980
  - Extension (Profile Toolbox) 560
  - Generalization 981
  - Generalize 981
  - Generalize (Profile Toolbox) 560
  - Hide 666
  - Implements 1007
  - Include 982
  - Information Flow 983
  - Inheritance 981
  - Interrupt Flow 985
  - List, On Context References Tab 686
  - Manifest 986
  - Matrix, In Traceability 494
  - Message 986
  - Mind Mapping 1182
  - Modify Using Relationship Matrix 500
  - Nesting 1002
  - Notelink 1003
  - Object Flow 1003
  - Occurrence 1005
  - Package Import 1006
  - Package Merge 1006
  - Pkg Import 1006
  - Pkg Merge 1006
  - Realize 1007
  - Recursion 1008
  - Redefinition 560
  - Representation 1010
  - Represents 1009
  - Role Binding 1008
  - Self Message 989
  - Show 666
  - SysML Activity 1553
  - SysML Block Definition 1556
  - SysML Interaction 1558
  - SysML Internal Block 1559
  - SysML Model 1561
  - SysML Parametric 1563
  - SysML Requirement 1564
  - SysML State Machine 1565
  - SysML Use Case 1567
  - Tagged Value (Profile Toolbox) 560
  - Trace 1012
  - Transition 1013
  - Usage 1014
  - Use 1014
  - Visibility 749
  - Window, Context Menu 504
- Relationship Matrix
  - Access 497
  - Access From Shortcut 140, 142
  - Access From Shortcut (Direct Definition) 141
  - Create Relationship 500
  - Delete Relationship 500
  - Export to CSV 501
  - Incorporate In HTML Report 501
  - Incorporate In RTF Report 501
  - Introduction 496
  - Link Direction 498
  - Link Type 498
  - Locate Elements 503
  - Manage Display Content 501
  - Modify Relationship 500
  - Open 497
  - Options 501
  - Print Matrix 501

- Relationship Matrix
  - Print Preview 501
  - Profile, Delete 499
  - Profile, Save 499
  - Profile, Update 499
  - Review Elements 503
  - Save As .EMF File 501
  - Save As .PNG File 501
  - Scale Printout Width 501
  - Set Element Type 497
  - Set Link Direction 498
  - Set Link Type 498
  - Set Source Package 498
  - Set Target Package 498
- Relationships Window
  - Context Menu 580
- Reload
  - Diagram 187
  - Model (Shared) 187
  - Project 187
  - View 187
- Reload Current Project
  - Menu Option (File Menu) 74
- ReloadType Enumeration
  - Automation Interface 1837
- Remove
  - Package Control 331
  - Recent Project 71
  - Replication 192
- REMOVE\_PREFIX
  - Macro 1341
- Rename
  - Diagram 581
  - Package 534
  - Project .EAP File 418
  - Views 531
- Re-Order
  - Messages 998
- Reorder Columns 1352
- Repair
  - Project .EAP File 419
- Replace
  - Linked Document 733
- Replica
  - Create 190
  - Synchronize 191
  - Upgrade 193, 417
- Replication
  - And Version Control 275
  - Change Collisions 188
  - Create Design Masters 190
  - Create Replicas 190
  - Disable 188
  - Impact On Auto Increment Columns 416
  - Introduction 188
  - Merge Rules 188
  - Remove 192
  - Resolve Conflicts 193
  - Synchronize Replicas 191
  - Upgrade Replicas 193
  - Using 188
- Reply
  - To Post In Team Review 224
- Report
  - Dependency 1783
  - HTML 1724, 1802
  - Implementation 1785
  - Open In Microsoft Word 1796
  - Project Issues, Via Project Issues Dialog 360
  - Project Issues, Via Project Issues Tab 360
  - Rich Text Format 1724, 1726
  - RTF 1724, 1726
  - Testing Details 1786
- Reporting
  - Dependency 1173
  - Implementation 1173
  - Requirements 1173
- Repository
  - Access Permissions For 148
  - Adaptive Server Anywhere, Create 152
  - Attributes 1855
  - Author Collection 1839
  - Automation Interface 1855
  - Client Collection 1840
  - Collection Class 1841
  - Connect To 148
  - Create 148
  - Datatype 1843
  - Encrypt Password 140, 143
  - EventProperties 1844
  - EventProperty 1844
  - Methods 1855
  - ModelWatcher 1845
  - MSDE Server, Create 152
  - Open, Automation Interface Code Example 1953
  - Package 1845
  - Package, Automation Interface 1838
  - Progress OpenEdge, Create 153
  - ProjectIssues 1851
  - ProjectResource 1852
  - PropertyRole 1853
  - PropertyType 1853
  - Reference 1854

- Repository
  - Set Up Database 148
  - Stereotype 1871
  - Task 1872
  - Term 1873
  - Transfer Data Between 343
  - Use Extras, Automation Interface Code Example 1961
- Representation
  - Connector 1010
  - Relationship 1010
- Represents
  - Connector 1009
  - Relationship 1009
- Require User Lock
  - Apply User Lock 213
  - Release User Lock 213
- Require User Lock Policy 197
- Requirement
  - Aggregation 1161
  - Analysis 521
  - And Level Numbering 1164
  - And Use Cases 1164
  - Auditing 1171
  - Automation Interface, Element Package 1890
  - Autonumbering 1156
  - Baselines 1171
  - Change Management 1171
  - Changes 1171
  - Color Code Status 1159
  - Connect On Diagram 1161
  - Connect Through Relationship Matrix 1161
  - Connectors 1161
  - Convert From Responsibility 1167
  - Create From Text File Text 1156
  - Create In Diagram 1156
  - Create In Project Browser 1156
  - Create SysML Model 1572
  - Dependency Report 1173
  - Docked Windows 1169
  - Element 1154, 1164
  - Element Template 1160
  - Element, Hide/Show Connectors 751
  - Elements And Connectors 563
  - Extend Properties, Default Format 1160
  - External 665, 1164
  - Fast Generate Realization Connector 1161
  - Functional 664
  - Gather 1152
  - Group, Toolbox 563
  - Hide Stereotype Letter 1154, 1156
  - Hierarchies 1164
  - Implementation Report 1173
  - Import Via CSV 1162
  - Inherited, Show 585
  - Internal 664, 1167
  - Internal, Import As Test 1706
  - Internal, In Scenarios & Requirements Window 689
  - Issues 1171
  - Manage 1169
  - Model 1152, 1164
  - Model Template 521
  - Model Views 1171
  - Modeling 521
  - Non-Functional 664
  - Profile 1160
  - Properties, Display On Diagram 1161
  - Properties, Extended 1160
  - Properties, Standard 1157
  - Realization 1161
  - Report Template 1173
  - Reporting 1173
  - Requirements Management 521
  - Review 1169
  - Show Stereotype Letter 1154, 1156
  - Stability 664
  - Status, Color Coded 1159
  - Tagged Values 1160
  - Tagged Values, Display 1161
  - Template 521
  - Trace Through Connectors 1161
  - Trace Use Of 1170
  - User-Defined Attributes 1160
  - View 1170
  - What Is A? 1152
  - Windows For Tracing Use 1170
  - Windows For Viewing 1170
- Requirement Type
  - Define 785
- Requirements Diagram
  - Description 1174
  - Elements And Connectors 1174
  - Example 1174
- Requirements Management
  - And Enterprise Architect 1152
  - Enterprise Architect 7
  - In Example Model 1152
  - Overview 1152
- Requirements Model
  - SysML, Create 1572
- Requirements Modeling
  - Enterprise Architect 7
- Reserved Names

- Reserved Names
  - In Shape Scripts, Connectors 1103
  - In Shape Scripts, Elements 1103
- Reset
  - Auto Increment Column Values 416
- Reset Options
  - For A Class 1540
  - For All Classes 1540
  - Source Code Language 452, 1540
- Reset Sort Order
  - In Project Browser 44
- Resize
  - Element 636
  - Element By Increments 636
  - Multiple Elements 636
- Resolve Change Conflicts
  - Between Replicas 193
- Resource
  - Add To Team Review Item 218
  - Allocation 349
  - And Tasking Details Dialog 370
  - Automation Interface, Element Package 1891
  - Delete From Team Review Item 218
  - On Task Element 1299
  - Report 370
- Resource Allocation
  - In Project Calendar 396
  - Project Task Allocation Window, Resource View 366
  - Record Work, Personal Information Window 382
- Resource Document
  - RTF Generator (Enhanced) 1766
  - RTF, Batch Generate Reports 1766
- Resource Management 348
  - Effort Types 353
  - Metric Types 354
  - Risk Types 355
- Resource View
  - Project Task Allocation 366
- Resources
  - Define 781
  - Documents 789
  - Favorites Folder 789, 790
  - Matrix Profiles 789
  - MDG Technologies 789
  - Project Management Window 347
  - Scripts 789
  - Templates 789
  - UML Patterns 789
  - UML Profiles 789
  - Window 789
- Window, And MDG Technologies 1031
- XSL Stylesheets 789
- Resources View
  - Of Patterns 1022
- Responsibility
  - Compartment, Element 659
  - Define 1167
  - Import As Test 1706
  - Internal 664
  - Move To External Requirement 1167
- Result
  - End Event, BPEL 1247
- Resynchronize
  - Package Version Control Status 297
- Reusable Subsystems
  - Systems Engineering Modeling 1577
- Reverse Connector 753
- Reverse Engineer
  - Enterprise Architect 7
  - Source Code 1508
- Reverse Engineering
  - And Auditing 307
  - And MDG Integration 1514
  - Directory Structure 1513
  - Eclipse 1514
  - Handling Classes Not Found During Import 1515
  - Import ActionScript 1510
  - Import Binary Module 1513
  - Import C 1510
  - Import C# 1510
  - Import C++ 1510
  - Import Delphi 1510
  - Import Java 1510
  - Import PHP 1510
  - Import Python 1510
  - Import Source Code 1510, 1515
  - Import Visual Basic 1510
  - Import Visual Basic.Net 1510
  - Initial Code In Operations 711
  - Introduction 1386
  - ODBC Data Sources 1375
  - Source Code, Import Directory Structure 1513
  - Synchronize Model And Code 1506
  - Visual Studio 1514
- Reverse Synchronization
  - Delete Attribute If Not In Code 1522
  - Delete Method If Not In Code 1522
  - Delete Model Aggregations For Attributes Not In Code 1522
  - Delete Model Associations For Attributes Not In Code 1522

- Reverse Synchronization
  - Include Method Bodies In Model 1522
- Review
  - Package Version Control History (SCC) 294
  - Package Version Control History (TFS, CVS, Subversion) 293
  - Requirements 1169
- Rich Text Format
  - Copy Bookmark To Clipboard 447
  - Document 1726
  - Report 1726
- Rich Text Format Generator
  - Enhanced 1732
- Rich Text Format Report
  - Apply Filter (Legacy) 1789
  - Diagram Format (Legacy) 1790
  - Diagram Only 1784
  - Dialog (Legacy) 1787
  - Element-Level 1727
  - Exclude Elements (Legacy) 1790
  - Exclude Objects (Legacy) 1790
  - Exclude Package 1731
  - Generate (Enhanced) 1727
  - Generate (Legacy) 1792
  - Generate From Diagram List 462
  - Generate From Package Browser 456
  - Generate, Quick Start 1727
  - Generator (Legacy) 1787
  - Include Glossary (Legacy) 1790
  - Include Issues (Legacy) 1790
  - Include Package 1731
  - Include Tasks (Legacy) 1790
  - Object Selections (Legacy) 1791
  - Options (Legacy) 1790
  - Save As RTF Document (Enhanced) 1766
  - Save As RTF Document (Legacy) 1793
  - Set Main Properties (Legacy) 1788
  - Single Element (Legacy) 1788
  - Templates, Load (Legacy) 1792
  - Through Diagram List 1727
  - Through Model Search 1727
  - Through Package Browser 1727
  - Wizard (Legacy) 1787
- Rich Text Notes
  - In Legacy RTF Generator Reports 1787
- Rigorous Security Mode
  - Apply User Lock 213
  - Release User Lock 213
- Risk
  - Automation Interface, Element Package 1891
  - Element 1298
  - Management 351, 1298
  - Project Management Window 347
- Risk Types
  - Define 355
  - Global 355
  - Non-Global 351
- Roadmap
  - ICONIX 1542
- Robustness Diagram 1288
  - Generate From Scenario 676, 683
- Role
  - Context Menu 736
  - Define 780
  - Tagged Values 760
- Role Binding
  - Connector 1008
  - Relationship 1008
- RoleTag
  - Automation Interface, Connector Package 1919
- Rollback Change
  - Options 316
- Round-Trip Engineering
  - Introduction 1386
- RSA
  - Models 324
  - XMI 323
- RSM
  - Import \*.emx Files 323
  - Import \*.uml2 Files 323
- RTF
  - Copy Bookmark To Clipboard 447, 451
  - Documentation 1724, 1726
  - Documentation, Other 1783
  - List Overrides 1761
  - Numbered Sections 1761
  - Numbering Levels 1761
  - Numbering List 1761
  - Report 1724, 1726
  - Section Numbering 1761
- RTF Document Options
  - RTF Generator (Enhanced) 571
  - RTF Generator (Legacy) 571
- RTF Generator
  - Document Options (Enhanced) 571
  - Document Options (Legacy) 571
  - Document Options, From Diagram (Enhanced) 1729
  - Document Options, From Diagram (Legacy) 1729
  - Enhanced 1732
  - Report External Elements 1730
- RTF Report
  - Advanced Options 1767

- RTF Report
- Apply Filter (Legacy) 1789
  - Bookmarks 1796
  - Bookmarks, In Master Document Elements 1774
  - Custom Language Settings 1773
  - Custom Language Settings (Legacy) 1794
  - Diagram Format (Legacy) 1790
  - Diagram Only 1784
  - Dialog (Legacy) 1787
  - Document Options (Enhanced Generator) 1767
  - Element Filters 1770
  - Element-Level 1727
  - Exclude Elements (Legacy) 1790
  - Exclude Objects (Legacy) 1790
  - Exclude Package 1731
  - Footers, Add 1799
  - Generate (Enhanced) 1727
  - Generate (Legacy) 1787, 1792
  - Generate From Diagram List 462
  - Generate From Package Browser 456
  - Generate, Quick Start 1727
  - Headers, Add 1799
  - In MS Open Office 1767
  - In Sun Open Office 1732
  - Include Glossary (Legacy) 1790
  - Include Issues (Legacy) 1790
  - Include Package 1731
  - Include Tasks (Legacy) 1790
  - Keywords (Legacy) 1794
  - Object Selections (Legacy) 1791
  - Open Office Display 1767
  - Options (Legacy) 1790
  - Other Filters 1771
  - Project Constants, User Defined 1772
  - Quick Start 1727
  - Save As RTF Document (Enhanced) 1766
  - Save As RTF Document (Legacy) 1793
  - Set Main Properties (Legacy) 1788
  - Single Element (Legacy) 1788
  - Switch Generator 1767
  - Templates, Load (Legacy) 1792
  - Through Diagram List 1727
  - Through Model Search 1727
  - Through Package Browser 1727
  - Update Links In MS Word 1801
  - Wizard (Legacy) 1787
  - Word Substitution 1773
- RTF Report Template
- Add To MDG Technology 1072
- RTF Style Editor (Legacy) 1792
- RTF Style Template
- Create 1734
  - Delete 1734
  - Edit 1734
  - Export To Reference File 1734
  - Import From Reference File 1734
  - Normal.rtf 1734
- RTF Style Template Editor
- Add Content 1742
  - Bookmarks 1755
  - Character Formatting 1750
  - Character Styles 1750
  - Child Sections 1740
  - Columns 1759
  - Commands 1743
  - Constraint Sections 1741
  - Continuous Table 1738
  - Copy Text 1746
  - Cut And Paste 1746
  - Delete Text 1746
  - Description (Enhanced) 1736
  - Drawing Objects 1763
  - Edit Picture 1749
  - Export As RTF Document 1745
  - File Options 1745
  - Fonts 1750
  - Footers 1754
  - Frames 1763
  - Headers 1754
  - Highlight Text 1746
  - Hyperlinks 1755
  - IME Option 1759
  - Import RTF Document 1745
  - Insert RTF File 1749
  - Level Numbering 1759
  - Lists And Overrides 1759
  - Move Text 1746
  - Non-rendered Subsections 1740
  - Numbered Lists 1751
  - Page Breaks 1754
  - Page Columns 1759
  - Page Mode 1747
  - Paragraph Formatting 1751
  - Paragraph Numbering 1751
  - Paragraph Styles 1751
  - Paste External Objects 1746
  - Picture Embed 1749
  - Picture Frame 1763
  - Picture Link 1749
  - Print Options 1745
  - Redo Edit 1743
  - Rendered Subsections 1740
  - Repagination 1754

- RTF Style Template Editor
    - Revert To Previous Copy 1745
    - Save File 1745
    - Scenario Sections 1741
    - Search and Replace Options 1764
    - Sections 1759
    - Select Model Elements 1737
    - Select Text 1746
    - Show Headers & Footers 1747
    - Special Text 1759
    - Split Rows 1738
    - Style Sheets 1759
    - Tab Support (Enhanced) 1753
    - Table Commands 1756
    - Table of Contents 1759
    - Tabular Sections 1738
    - Text Frame 1763
    - Text Scrolling 1744
    - Undo Edit 1743
    - Update Styles 1745
    - ValueOf Field 1742
    - View Options 1747
    - Zoom 1747
  - RTF Template
    - Import 1765
  - RTF Templates Tab (Enhanced) 1734
  - Rule Composer
    - Business Rule Modeling 1211
    - Computation Rule Table 1211
    - Decision Table 1211
    - Rule Table 1211
  - Rule Flow
    - Activity 1206
    - Activity Parameter 1208
    - Behavior 1206
    - Diagram 1206
    - Generate Code From Behavior 1217
    - Model 1206
  - Rule Model
    - Diagram 1202
  - Rule Task
    - Element 1206
  - RuleFlow Diagram
    - Generate From Scenario 676, 678
  - Run
    - Application, Analyzer menu Option 94
    - SQL Patches 418
  - Run Command
    - Create 1451
    - Introduction 1451
  - Run Script
    - Create 1451
  - Introduction 1451
  - Run State 958
    - Add Instance Variable 958
    - Define, Element Context Menu 649
  - Run-Time
    - Variable, Define 958
  - Runtime Object
    - In Script Editor 1407
  - Run-Time State
    - Add Instance Variable 958
    - Delete Instance Variable 959
    - Introduction 958
- S -**
- Save 331
    - Changes 52
    - Controlled Package 331
    - Diagram As UML Pattern 1022
    - Diagram Changes Automatically 430
    - Diagram Image To Disk File 582
    - Diagram, Context Menu Option 538
    - Diagram, Quick Start 52
    - Learning Center Profile 1086
    - Package with XMI 331
    - Profile From Diagram Context 1050
    - Profile From Package Context 1050
    - RTF Report As RTF Document (Enhanced) 1766
    - RTF Report As RTF Document (Legacy) 1793
    - UML Pattern 1022
  - Save As
    - Shortcut 140
  - Save Project As
    - Menu Option (File Menu) 74
  - Save Project As (File Menu Option)
    - Copy 141, 142
    - Shortcut (Capture Current Environment) 142
    - Shortcut (Direct Definition) 141
  - Scale
    - Image To Page Size 602
  - Scale Diagram View 427
  - Scan XMI and Reconcile
    - Version Control 278
  - SCC
    - Package Version Control History, In 294
    - Providers Dialog 271
    - Version Control Options 271
  - Scenario
    - Alternate Path 667, 669
    - Alternate Path, Add 671, 672
    - Alternate Path, RTF Reporting 1741

- Scenario
  - Automation Interface, Element Package 1892
  - Basic Path 667
  - Context Menu, Item 672
  - Create 669, 671, 672
  - Create Element 674
  - Delete Path 675
  - Delete Step 671, 672
  - Delete Text 674
  - Description Tab 667
  - Edit Text 674
  - Element 667
  - Entry Points Tab 669
  - Entry Points Tab, Context Menu 675
  - Exception Path 667
  - Exception Path, 669
  - Exception Path, Add 671, 672
  - External Test Cases 685
  - Floating Toolbar 676
  - Generate Activity Diagram From 677
  - Generate Diagram From 676
  - Generate From Activity Diagram 684
  - Generate From Clipboard Text 672
  - Generate Robustness Diagram From 683
  - Generate RuleFlow Diagram From 678
  - Generate Sequence Diagram From 681
  - Generate State Machine Diagram From 679
  - Generate Structure From Notes 667
  - Glossary Reference, Add 674
  - In Scenarios & Requirements Window 689
  - Insert Context Reference 674
  - Internal Test Cases 685
  - Item Context Menu 672
  - Join Scenarios 675
  - Link Step To Use Case 672, 674
  - Link To Element 674
  - Merge Steps 672
  - Move Step 671
  - Move Steps 672
  - Object 667
  - Organization 675
  - RTF Report Template Sections 1741
  - Set Responsible Entity 672
  - Split Step 674
  - Step Numbers On Sequence Messages 681
  - Structured Specification Tab 667, 669
  - Structured Specification, Create 669
  - Structured, RTF Reporting 1741
  - Test Cases 685
  - Testing 1701
  - Text Context Menu 674
  - Toolbar 671
  - Type, Define 786
  - Undo Changes 674
  - Use Case 853
- ScenarioDiagramType Enum
  - Automation Interface 1837
- ScenarioExtension
  - Automation Interface, Element Package 1893
- Scenarios & Requirements
  - Toggle Window/View 689
  - Window 689
- ScenarioStep
  - Automation Interface, Element Package 1894
- ScenarioStepType Enum
  - Automation Interface 1838
- ScenarioTestType Enum
  - Automation Interface 1838
- Schema
  - Database 1278
  - Database, Import From ODBC 1375
  - DDL, Import From ODBC 1375
  - Owner Tagged Value 1349
  - Set Owner 1349
- Scope
  - Values 1457
- Scope Guides
  - Code Editor, Common 1398
- Screen
  - Element 1281
- Script
  - Add To MDG Technology 1072
  - Commands 1820
  - Console 1820
  - Copy 1818
  - Create 1818
  - Delete 1818
  - Deploy, Create 1453
  - Deploy, Introduction 1453
  - Diagram 1819
  - Engines 1817
  - Execute 1818
  - Group 1817, 1818
  - Group Type 1819
  - Hyperlink To 1294
  - JavaScript 1817, 1818
  - JScript 1817, 1818
  - Local 1817, 1818
  - Move 1818
  - Normal 1819
  - Project Browser 1819
  - Run, Introduction 1451
  - Search 1454, 1819
  - Template 1817, 1818

- Script
  - Unit Test, Create 1451
  - Unit Test, Introduction 1451
  - User 1817, 1818
  - VBScript 1817, 1818
  - Workflow 1819
  - Workflow Functions 232
  - Workflow, Introduction 231
  - Worklow 1818
- Script Actions
  - Define, Visual Execution Analyzer 1394
- Script Editor
  - Context Menu 1404
- Script Objects 1407
- Scripting
  - In Enterprise Architect 1816
- Scripting Window
  - Console Tab 1820
  - Console Toolbar 1820
  - Context Menu 1818
  - Create Script 1817
  - Editor 1407
  - Group Properties Dialog 1819
  - Intellisense 1407
  - Runtime Objects 1407
  - Script Editor 1407
  - Script Objects 1407
  - Scripts Tab 1818
  - Toolbar 1818
  - Type Libraries 1407
- Scripts
  - Build 1415
- SDK
  - Enterprise Architect 1038
- Search
  - Add-In 1974
  - Advanced Options 486
  - Code Editor Facility 1404
  - Conditions 489
  - Configure Code Editor Context Menu Options 1404
  - Debugger File Search 1454
  - Definitions 482
  - Definitions, New 484
  - Definitions, Predefined 479
  - Element Features 482
  - Element Features, RTF Reports 1771
  - Element Filters, RTF Reports 1770
  - Fields 489
  - File, Introduction 1454
  - Filters 482
  - List, Predefined Searches 479
  - Macros, SQL 484
  - Manage 482
  - Model 482
  - Model, Create Search Definition 484
  - Other Filters, RTF Reports 1771
  - Project 482
  - Query, New 484
  - Results, Manipulate 475
  - Scripts 1454, 1819
  - Simple 482
  - Team Review 229
  - User Defined, Storage 484
  - Within Project Browser 452
- Search Data Parameter
  - Add-In Search 1975
- Search Filters
  - Add To Search 487
- Search Project
  - Add Filters 487
- Search Tools 440
- Section Numbering
  - In RTF Reports 1761
  - In Virtual Documents 1782
- Security
  - Basics 194
  - Change Password 198, 209
  - Disable 196
  - Enable 196
  - Maintain Groups 203
  - Maintain Users 198
  - Policy 197
  - Re-enable 196
  - Require User Lock Mode 197
  - Reset Password 209
  - Rigorous Security 197
  - Set Password 209
  - Standard Security 197
  - Submenu (Project Menu) 82
  - Tasks 194
  - User Permission List 205
  - User/Group Lock Mode 197
  - What Is User Security? 194
- Security Group Permissions 204
- see the Color Code External Requirements topic 1156
- See the Pass Parameters to Rule Flow Activity 1206
- Select
  - Alternative Image 593
  - ODBC Data Source 1377
  - Stereotypes 1017
- Select <Item> Dialog 690

- Select All
  - Menu Option (Edit Menu) 74
- Select Attribute Type
  - Data Type 696
  - Dialog 696
- Select By Type
  - Menu Option (Edit Menu) 74
- Select Property Dialog 692
- SELECT Statement
  - SQL Search 484
- Selectable
  - Element 655
- Self Message
  - Calls 991
- Self-Message
  - Connector 989
  - Hierarchy, Sequence Diagram 856
  - Relationship 989
  - Return 989
- Send
  - Element 917
  - Event 1292
- SendSignal Action
  - Signal Tab 866
- Sent Mail
  - Personal Information Window 378
- Sequence
  - Communication Messages 998
  - Elements and Connectors 556
  - Message, Change Timing Details 993
  - Message, Create 987
  - Message, Timing Details 993
  - Oracle, DDL Options 1379
  - Oracle, DDL Options For Packages 1380
- Sequence Diagram
  - Activation Levels 856
  - And Version Control 854
  - Damage To 854
  - Description 849
  - Diagram Features, Generate Sequence Diagrams 1652
  - Display Options 431
  - Element Activation 855
  - Elements 853
  - Elements And Connectors 849
  - Example 851
  - Generate Code From 1496, 1503
  - Generate From Debugger Call Stack 1450
  - Generate From Recording, Execution Analysis 1651
  - Generate From Scenario 676, 681
  - Generate In Execution Analysis 1638
  - Generate, Execution Analysis 1652
  - Layout 852
  - Lifeline Activation Level 856
  - Messages, Asynchronous Signal 996
  - Messages, Self Message 989
  - Recording History, Execution Analysis 1651
  - Save Recording History, Execution Analysis 1653
  - Self-Message Hierarchy 856
  - Step Numbers On Messages 681
  - Top Margin, Change 858
- Sequence Diagram Message
  - Examples 992
  - External To Sequence 992
  - General Ordering 994
- Sequence Diagrams
  - Filter calls 1639
- Sequence Element
  - Inline, Part And Port 858
- Sequence Flow
  - Create 1248
  - Model In BPEL 1248
- Sequence Message
  - Label Visibility 858
  - Modify Height 852
- Sequence Recording Option
  - Record Activity For Class 1646
  - Record Activity For Method 1645
- Sequential Node
  - Structured Activity 926, 927
- Server
  - Apache Tomcat, Debugging 1425
  - JBOSS, Debugging 1425
  - Tomcat, Debugging 1425
- Server Configuration
  - JBOSS 1427
  - Tomcat 1428
- Server Repository
  - Create for Oracle 10g 151
  - Create for Oracle 9i 151
- Service
  - WSDL Diagram 1619
  - WSDL Element 1619
- Service Configuration
  - Tomcat 1428
- Service Oriented Architecture
  - Development 1389
  - Implementing XML-Based, In Enterprise Architect 1389
- Service Oriented Architecture Modeling Language 1624
- Service Oriented Modeling Framework 1628

- Set
  - Activities For Transitions 690
  - Association Specialization 753
  - Classifiers For REFGUID Tagged Values 690
  - Collection Classes 1526
  - Connector Visibility 749
  - Default Diagram, Model 585
  - Default Tree Behavior 455
  - Diagram Appearance Options 569
  - Diagram Page Size 600
  - Diagram Properties 569
  - Element Classifier 1024
  - Element Cross References 630
  - Element Custom References 630
  - Element Parent 628
  - Feature Visibility 585
  - Font For Element Text 657
  - Group Permissions 204
  - Instance Classifier 690
  - Interface For Element 628
  - Main RTF Report Properties (Legacy) 1788
  - Message Source and Target 754
  - Object State 959
  - Operation Parameter Return Type 690
  - Operation Parameter Type 690
  - Operation Return Type 690
  - Parent For Element 628
  - Pattern Element Defaults 690
  - Relationship Visibility 749
- Set Attribute Dialog 694
- Set Element Project Colors 425
- Set Feature Dialog 694
- Set Feature Visibility
  - Element Context Menu Option 654
- Set Function
  - Create As Attribute Property 700
- Set Operation Dialog 694
- Set Project Custom Colors
  - For Element 642
- Set Up
  - Adaptive Server Anywhere ODBC Driver 158
  - Database Model Files 148
  - Debug Session 1417
  - For .NET 1429
  - MySQL ODBC Driver 154
  - ODBC Driver 154
  - Oracle ODBC Driver 160
  - PostgreSQL ODBC Driver 155
  - Progress OpenEdge ODBC Driver 159
  - Single Permissions 201
  - User Groups 201
- Settings
  - Author 778
  - Cardinality (Multiplicity) 776
  - Clients 782
  - Constraint Status Types 785
  - Constraint Types 784
  - Default Hours 404
  - Effort Types 353
  - Environment Complexity Factors 402
  - Estimation 787
  - General Types 782
  - Maintenance 787
  - Menu 772
  - Menu, Configure 98
  - Metric Types 354
  - Metrics 787
  - People 778, 781
  - Problem Types 787
  - Project Author 778
  - Project Resources 781
  - Requirement Types 785
  - Risk Types 355
  - Roles 780
  - Scenario Types 786
  - Status Types 783
  - Stereotypes 773
  - Tagged Value Types 775
  - Technical Complexity Factors 401
  - Template Package 643
  - Testing Types 788
  - UML Types 772
- Shallow Copy
  - Of Diagram 583
- Shallow History 902
  - Change To Deep 649
- Shape
  - <LabelID> 1103
  - Attributes 1092
  - Decoration 1103
  - Editor 1090
  - Label 1103
  - Main 1103
  - Source 1103
  - Target 1103
- Shape Attributes
  - Shape Scripts 1092
- Shape Editor 774, 1090
- Shape Scripts
  - Add To Profile 1048
  - Arithmetical Operations 1103
  - Assign To Stereotype 1090
  - Basic Shapes 1104
  - Change Font Of Text 1103

- Shape Scripts
  - Cloud Path 1104
  - Color Queries 1099
  - Comments 1103
  - Conditional Branching 1099
  - Connector 1104
  - Create 1090
  - Custom Shapes 1089
  - Display Element Properties 1099
  - Double Line 1104
  - Drawing Methods 1094
  - Editable Field 1104
  - Example Shape Scripts 1104
  - Filled Arrow 1104
  - Fonts 1103
  - Getting Started 1090
  - Introduction 1089
  - Looping 1103
  - Miscellaneous 1103
  - Multiple Condition 1104
  - Override Element Appearance 1090
  - Properties, Connector 1099
  - Properties, Element 1099
  - Query Methods 1099
  - Reserved Names, Connectors 1103
  - Reserved Names, Elements 1103
  - Return Statement 1104
  - Shape Attributes 1092
  - Shape Editor 774, 1090
  - Single Condition 1104
  - Stereotypes 1089
  - String Manipulation 1103
  - Subshape 1104
  - Subshape Layout 1102
  - Syntax Grammar 1091
  - Terminate Execution 1103
  - Variable Declarations 1103
  - Without Stereotypes 1103
  - Writing Scripts 1091
- Share
  - An Enterprise Architect Project 186
  - Project On Network Drive 187
- Shared Key
  - Add 2096
  - Issues 2098
- Shared Model Development
  - Enterprise Architect 7
- Shortcut
  - Clear 142
  - Diagram (Capture Current Environment) 142
  - Keyboard 128
  - Keyboard/Mouse 133
  - Menu, Toolbox 551
  - Model (Capture Current Environment) 142
  - Model Search (Capture Current Environment) 142
  - Relationship Matrix (Capture Current Environment) 142
  - Team Review (Capture Current Environment) 142
- Shortcut To
  - Diagram 140
  - Diagram (Direct Definition) 141
  - Model 140
  - Model (Direct Definition) 141
  - Model Search 140
  - Model Search (Direct Definition) 141
  - Relationship Matrix 140
  - Relationship Matrix (Direct Definition) 141
  - Team Review 140
  - Team Review (Direct Definition) 141
- Show
  - Connectors, All Diagrams 751
  - Connectors, Requirements Element 751
  - Connectors, Single Diagram 751
  - Diagram Caption Bar 78
  - Duplicate Tagged Values 766
  - Element Stereotype 1018
  - Feature Stereotype 1018
  - Labels 752
  - Package Contents On Diagram 537
  - Project Browser 441
  - Relationship 666
  - Toolbox 546
  - Usage Of Element 630
  - Use Case Arrowhead 754
- Show Grid
  - Diagram Menu Option 83
- Show Status Colors on Diagrams
  - Menu Option 1159
- Show/Hide
  - Connectors 740
  - Labels 740
- Signal
  - Element 964
  - Reception 964
- Simple View 530
- Simulate
  - SysML Parametric Model 1570
- Simulate the parametric models to verify their 1548
- Simulation Script
  - Activate 1682
  - Run 1683
  - Set Up 1681

- Single Permissions
  - Set Up 201
- Single User 1524
- Size
  - Elements By Increments 88
  - Submenu 88
- Slideshow
  - Diagram Presentations 464, 471
  - Model View 464, 471
  - Of Diagrams, Automate 471
  - Properties Dialog, Model View 471
  - Remove From Model View 471
  - Run, Diagram View 471
  - Run, Full Screen View 471
  - Stop 471
- Snap to Grid
  - Diagram Menu Option 83
- Snapshot
  - Of Variables In Code Execution 1444
- SOA
  - Development 1389
  - Implementing XML-Based, In Enterprise Architect 1389
  - Modeling Language 1624
- SOA WSDL
  - Elements And Connectors, Enterprise Architect 565
- SOA XSD
  - Elements and Connectors, Enterprise Architect 566
- SoaML
  - Concept 1624
  - Diagrams 1624
  - Disable 1624
  - Elements 1624
  - Enable 1624
  - MDG Technology 1624
  - Toolbox Pages 1624
- SOAP Binding 1617
- Software and Systems Process Engineering Meta-model
  - SPEM 406
- Software Architect
  - And Enterprise Architect 59
  - Project Role 59
- Software Development
  - Database Engineering 1345
  - MDA Transformations 1301
  - Model Transformations 1301
  - Software Engineering 1386
  - XML Engineering 1580
- Software Development Kit
  - Enterprise Architect 1038
- Software Engineer
  - And Enterprise Architect 60
  - Project Role 60
- Software Product License Agreement 20
- SOMF
  - Concept 1628
  - Diagrams 1628
  - Disable 1628
  - Elements 1628
  - Enable 1628
  - MDG Technology 1628
  - Toolbox Pages 1628
- Sort Order
  - Project Browser, Reset 44
- Sorted Lookup Table
  - Create In Data Modeling 1366
- Source
  - Set for Message 754
- Source Code
  - Add New Features And Elements 1488
  - Control 242
  - Display In Source Code Viewer 1408
  - Display Parameter Information 1410
  - Editor 1408
  - Editor Functions 1410
  - Engineering, Project Browser Options 452
  - File Parsing In Source Code Viewer 1408
  - Generate For Method In Project Browser 452
  - Generate For Operation In Project Browser 452
  - Import, Reverse Engineering 1510
  - Internal Editor Options 1519
  - Open Directory, Project Browser Menu Option 448
  - Record Macro 1410
  - Reset Language 452, 1540
  - Reverse Engineer 1508
  - Synchronize 1508
  - Synchronize With Method In Project Browser 452
  - Synchronize With Operation In Project Browser 452
  - View For Method In Project Browser 452
  - View For Operation In Project Browser 452
  - View, Project Browser Menu Option 448
  - Viewer 1408
  - Viewer Toolbar 1410
  - XML Structure Tree 1408
- Source Code Engineering
  - Submenu (Element Menu) 88
  - Submenu (Tools Menu) 91
- Source Code Generation

- Source Code Generation
  - Class 1490
  - Interface 1490
  - Overview 1490
- Source File
  - Open (Option) 90
- Source Object
  - Multiplicity 758
- Source Role
  - Details 758
- Space Evenly
  - Submenu 88
- Sparx Systems
  - Enterprise Architect Community Site 30, 70
  - Website 30
- Specialize Association 753
- Specialized UML Models 1178
- Spell Check
  - Correct Words 377
  - Dictionary 377
  - Languages Other Than English 373
  - Model 376
  - Options, Configure 373
  - Perform 376
  - Project 376
  - Single Package 376
- Spell Checking
  - Automatic, Disable 372
  - Automatic, Enable 372
  - Introduction 372
- Spelling Options
  - Select 373
- SPEM
  - Base Plug-In 406
  - Concept 406
  - Diagram 407
  - Disable 406
  - Elements 407
  - Enable 406
  - MDG Technology 406
  - Method Content 406
  - Package 406
  - Process 406
  - Software and Systems Process Engineering Meta-model 406
  - Stereotype Presentation 407
  - Toolbox Page 407
- Spring Diagram Layout 613
- SQL
  - Custom Searches 484
  - Editor 484
  - Patches, Run 418
- Search Macros 484
- Search, SELECT Statements 484
- Server Data Repository, Connect To 163
- Server Repository, Create 150
- SQL Server
  - Default Constraint 1373
  - Desktop Engine, Upsize To 182
  - Non-Clustered Primary Key 1360
  - Upsize To 176
- Standard Colors
  - Attribute 424
  - Connector Line 424
  - Diagram Background 424
  - Element Fill 424
  - Element Line 424
  - Method 424
  - Note Text 424
  - Operation 424
  - Options 424
  - Screen 424
  - Set Element Project Colors 425
  - Shadow 424
- Standard Element Stereotypes 1019
- Start
  - Application 35
  - Enterprise Architect 35
- Start Event, BPEL
  - Create 1240
  - Model 1240
  - Types 1240
- Start Page 70
  - Hide 100, 543
  - Quick Start 37
  - Show 100
  - Show When Hidden 540
- State
  - Chart 815
  - Composite 917, 919
  - Diagram 815
  - Element 917
  - Entry And Exit Actions 708, 917
  - In Timeline Element 840
  - Locate In State Machine Diagram 827
  - Locate In State Machine Table 827
  - Operation Behavior 710
  - Reposition In State Machine Table 826
  - Simple 917
  - State Machine Table Conventions 828
- State (Machine)
  - Elements and Connectors 558
  - Group, Toolbox 558
- State Changes

- State Changes
  - Capture, Execution Analysis 1653
  - Map, Visual Execution Analyzer 1657
  - Record, Visual Execution Analyzer 1657
  - Set Up To Capture, Execution Analysis 1653
- State Invariant
  - Element 920, 922
- State Lifeline
  - Element 923
- State Lifeline Element
  - Add State 834
  - Add To Timing Diagram 833
  - Add Transition 836
  - Change Transition Time 835
  - Define Name 833
  - Delete State 835
  - Delete Transition 835
  - Edit State 835
  - Edit Transition 835
  - Merge Transitions 835
  - Move Transition 836
  - Set Timeline Start Position 833
  - Sizing and Scale 833
  - Synchronize Transition 835
- State Machine 1650
  - Element 925
  - Entry And Exit Actions 708
  - Model For Hardware Description Languages 1502
  - Regions 820
- State Machine Diagram 1655
  - Code Generated From 1498
  - Description 815
  - Display Format 815
  - Elements And Connectors 815
  - Example 817
  - Execution Analysis 1655
  - Generate Code From 1496, 1497
  - Generate From Scenario 676, 679
  - In Visual Execution Analyzer 1655
  - Locate State In State Machine Table 827
  - Locate Transition In State Machine Table 827
  - Locate Trigger In State Machine Table 827
- State Machine Table
  - Add States 824
  - Add Substates 824
  - Add Triggers 825
  - Cell Color 822
  - Cell Enumeration 822
  - Cell Highlights 822
  - Cell Size 822
  - Change Position In Diagram View 824
  - Change Size 824
  - ChangeTransitions 825
  - Conventions 828
  - Description 821
  - Export To CSV 828
  - Format 821
  - Insert Transitions 825
  - Legend, Add 826
  - Legend, Remove 826
  - Locate State In State Machine Diagram 827
  - Locate Transition In State Machine Diagram 827
  - Locate Trigger In State Machine Diagram 827
  - Operations, Overview 823
  - Options 822
  - Remove Substate Parent Relation 824
  - Reposition States 826
  - Reposition Substates 826
  - Reposition Triggers 826
  - State-Next State 821, 830
  - State-Trigger 821, 829
  - Table Format 822
  - Trigger-State 821
- State Region
  - Composite 820
- State Transitions 1650
  - Add, Visual Execution Analyzer 1653
- State/Continuation
  - Element 920
- Statement
  - Trace 1440
- Status Bar
  - Element Coordinates 117
  - Enterprise Architect Workspace 117
  - Hide 117
  - Show 117
  - Zoom Control 117
- Status Type
  - Color 783
  - Define 783
  - For Different Elements 783
- Step Into
  - Function Calls 1449
- Step Out Of
  - Functions 1449
- Step Over
  - Lines Of Code 1448
- Step Through
  - Function Calls 1449
- Stereotype
  - Add Shape Script In Profile 1048
  - Add To Profile 1042

- Stereotype
  - Add, Automation Interface Code Example 1963
  - Analysis 1186
  - And Metafiles 1016
  - Apply To Dependency Relationship 979
  - Automation Interface Repository 1871
  - Custom 1039
  - Define As Metatype 1054
  - Definition 1016
  - Dialog 1039
  - Extension 1288
  - Inbuilt 1288
  - Multiple, Restrict Application Of 1054
  - Predefined Tag Types 1045
  - Profile 1058
  - Selector 1017
  - Set Default Appearance Of Objects In Profile 1049
  - Settings 773
  - Show On Project Browser 455
  - SPEM Presentation 407
  - Standard Element 1019
  - Synchronize Element With Profile 1029
  - Tagged Values In Profile 1044
  - Tags For Supported Attributes 1045
  - Tags, Define 1044
  - UML Description 1016
  - Visibility 1018
  - With Alternative Images 1020
  - XSD In UML Profile 1602
- Stereotyped Element
  - Table 1277
- Store
  - Image In Enterprise Architect 593
- Stored Procedure
  - As Individual Class 1367
  - Definition 1366
  - Element 1367
  - Select From ODBC Data Source 1377
  - Supported Databases 1366
- Straighten
  - Line At Cursor 745
- Strategic Modeling
  - Integrated Technology 1036
- String Viewer Dialog 1443
- Structural Diagram
  - Elements 940
- Structural Diagrams
  - Overview 796
- Structural Specification
  - Generate From Description 667
- StructuralFeature Action
  - Set Structural Feature 866
- Structured Activity
  - Conditional Node 926, 929
  - Element 926, 927, 928, 929
  - Loop Node 926, 928
  - Nested 926
  - Node 926, 927
  - Sequential Node 926, 927
- Structured Activity Node 927
- Structured Scenario
  - Constraints Tab 686
  - RTF Report Template Section 1741
- Structured Specification
  - Alternate Path 669
  - Alternate Path, Add 671, 672
  - Context Menu, Item 672
  - Create 669, 671, 672
  - Create Element 674
  - Delete Path 675
  - Delete Step 671, 672
  - Delete Text 674
  - Edit Text 674
  - Entry Points Tab 669
  - Entry Points Tab, Context Menu 675
  - Exception Path 669
  - Exception Path, Add 671, 672
  - External Test Cases 685
  - Floating Toolbar 676
  - Generate From Activity Diagram 684
  - Generate From Clipboard Text 672
  - Glossary Reference, Add 674
  - Insert Context Reference 674
  - Internal Test Cases 685
  - Item Context Menu 672
  - Join Scenarios 675
  - Link Step To Use Case 672, 674
  - Link To Element 674
  - Merge Steps 672
  - Move Steps 672
  - Of Scenario Steps 669
  - Organization 675
  - Set Responsible Entity 672
  - Split Step 674
  - Test Cases 685
  - Text Context Menu 674
  - Toolbar 671
  - Undo Changes 674
- Structured Tagged Value Type 1108
- Structured Tagged Values
  - Create 1111
- Style
  - For Connectors 739

- Style Template
  - Fragments, HTML 1806
- StyleEx
  - Diagram Profile Attribute Values 1081
- Sub Activity
  - As Hyperlink 1294
- Sub-Activity
  - Conditional Node 926, 929
  - Element 873, 926, 927
  - Loop Node 926, 928
- Submachine State
  - Element 917, 925
- Submenu
  - Add-In 97
  - Advanced (Element) 86
  - Alignment 88
  - Appearance (Element) 88
  - Data Management (Tools Menu) 93
  - Database Engineering (Tools Menu) 92
  - Debug (Analyzer) 95
  - Documentation (Project Menu) 80
  - Hidden, Create In Toolbox Profile 1075
  - Import/Export (Project Menu) 81
  - Inline Features (Element Menu) 87
  - Make Same 88
  - Manage .EAP File (Data Management Menu) 94
  - Model Transformations (Tools Menu) 93
  - Model Validation (Project Menu) 81
  - More Analyzer Windows 97
  - More Element Tools 78
  - More Project Tools 78
  - Move 88
  - Online Resources 101
  - Paste Elements (Edit Menu) 75
  - Project Types 99
  - Record (Analyzer) 96
  - Security (Project Menu) 82
  - Size 88
  - Source Code Engineering (Element Menu) 88
  - Source Code Engineering (Tools Menu) 91
  - Space Evenly 88
  - Test (Analyzer) 97
  - Toolbars 78
  - Version Control (Project Menu) 82
  - Visual Styles 78
  - Web Services (Tools Menu) 92
  - XML Schema (Tools Menu) 92
  - Zoom 599
  - Z-Order 88
- Subsetting
  - Property, Port 963
- Subsetting Property
  - Of Attribute 698
- Subshape
  - Example 1102
  - In Shape Scripts 1102
- Substate
  - Reposition In State Machine Table 826
- Sub-State 919
- Substitute Words
  - In Extended RTF Generator 1773
  - In RTF Report (Legacy) 1794
- Substitution
  - Conditional 1116
  - Direct 1116
  - Macro 1099
- Subtype
  - Relationship 750
- Subversion
  - Package Version Control History, In 293
  - TortoiseSVN 260
  - Using With Enterprise Architect Under WINE Crossover 258
  - Version Control Options 273
  - Version Control, Create Local Working Copy 256
  - Version Control, Create Repository Subtree 256
  - Windows-Based Client 258
- Support
  - For Registered Users 31
  - For Trial Users 31
- Supported Attribute
  - Create Composite Elements 1056
  - Define Behavior On Creating Instance 1055
  - Define Child Diagram Types 1057
  - Metatype, In Profiles 1051
  - Of Stereotype Tags 1045
  - Stereotype, In Profiles 1051
- Supported Stereotype Attribute Tags 1045
- Suppress
  - Line Segments 745
- SwimlaneDef
  - Automation Interface, Diagram Package 1927
- Swimlanes
  - Automation Interface, Diagram Package 1927, 1928
  - Manage 590
  - On Diagram 590
  - Orientation 590
- Swimlanes Matrix
  - Activate 592
  - And Matrix Dialog, Matrix Tab 592
  - Create Columns And Rows 592

- Swimlanes Matrix
  - Define Heading 592
  - Delete Items 592
  - Edit Items 592
  - Lock 592
  - Model Profile 592
  - Size 592
- Switch
  - Between Diagram Tabs 545
  - Diagrams 540
  - RTF Generator 1767
- Sybase Adaptive Server Anywhere
  - ODBC Driver, Set Up 158
  - Upsize To 180
- Synch
  - Element 930
- Synchronization 188
  - Initial Code In Operations 711
  - Introduction 1386
  - Macros, Code Template Syntax 1133
  - Of Source Code And Model 1508
- Synchronize
  - Batch With Code 88
  - By Dragging Element From Toolbox 1029
  - Call Argument With Behavior Parameter 716
  - Class With Code 88
  - Classes, Import Database Schema 1375
  - Code 1486
  - Elements With Profile 1029
  - Existing Code Sections 1487
  - Invocation Argument With Behavior Parameter 716
  - Package With Source, (Tools Menu Option) 91
  - Replicas 191
  - Stereotypes From Profile 1029
  - Tagged Values And Constraints 1029
  - Tagged Values From MDG Toolbox Pages 1029
  - UML Profile Tagged Values And Constraints 1029
- Syntax Check
  - For UML, Option 426
- Syntax Grammar
  - Shape Scripts 1091
- Syntax Highlighting
  - Code Editor Options 1520
  - Code Editor, Common 1395
- SysML
  - Activity Elements, Toolbox Page 1553
  - Activity Relationships, Toolbox Page 1553
  - Block Definition Elements, Toolbox Page 1556
  - Block Relationships, Toolbox Page 1556
- Concepts 1552
- Create Operational Domain Model 1573
- Design Model 1575
- Diagrams 1552
- Disable 1552
- Interaction Elements, Toolbox Page 1558
- Interaction Relationships, Toolbox Page 1558
- Internal Block Diagram 1575
- Internal Block Elements, Toolbox Page 1559
- Internal Block Relationships, Toolbox Page 1559
- MDG Technology 1552
- MDG Technology For, Enterprise Architect 1036
- Migrate 1.1 Model To 1.2 1578
- Model Elements, Toolbox Page 1561
- Model Relationships, Toolbox Page 1561
- Parametric Elements, Toolbox Page 1563
- Parametric Model, Simulate 1570
- Parametric Models 1568
- Parametric Relationships, Toolbox Page 1563
- Requirement Elements, Toolbox Page 1564
- Requirement Extensions, Toolbox Page 1564
- Requirement Relationships, Toolbox Page 1564
- Requirements Model, Create 1572
- Reusable Subsystems 1577
- State Machine Elements, Toolbox Page 1565
- State Machine Relationships, Toolbox Page 1565
- Toolbox Pages 1552
- Use Case Elements, Toolbox Page 1567
- Use Case Relationships, Toolbox Page 1567
- System
  - Project Glossary 356
  - Project Issues 356
  - Project Tasks 356
  - Tabs 356
  - Testing 1699
  - Users 198
  - Window 356
- System Boundary
  - Element 931
  - Insert New From Toolbar 110
- System Design,
  - Compose, Systems Engineering Modeling 1575
- System Engineering Modeling
  - Create Reusable Subsystems 1577
- System Window
  - Project Glossary Tab 362
  - Project Issues Tab 358
  - Project Tasks Tab 356
- SystemC

- SystemC
    - Code Generation 1537
    - Enterprise Architect Toolbox Pages 1474
    - Language Options 1537
    - Modeling Conventions 1474
  - Systems Engineering Edition
    - Of Enterprise Architect 14
  - Systems Engineering Model
    - Create 1548
  - Systems Engineering Modeling
    - Compose System Design 1575
    - Create SysML Parametric Model 1568
    - Create SysML Requirements Model 1572
    - Overview 1548
    - Process 1548
    - Simulate SysML Parametric Model 1570
    - SysLM Operational Domain Model, Create 1573
  - Systems Modeling Languages
    - MDG Technology For, Enterprise Architect 1036
  - Systems Modelling Language (SysML) 1552
- T -
- Tab
    - Asterisks 543
    - Audit History In Output Window 127
    - Close 543
    - Context Menu 543
    - Context References 686
    - Diagram 543
    - View 543
  - Tab Support
    - RTF Style Template Editor (Enhanced) 1753
  - Tabbed Frame
    - Combine Windows In 103
    - Remove Window From 103
  - Table
    - Create 1347
    - Data Modeling 1347
    - DDL Script For 1276
    - Detail 1277
    - Element 1277
    - Introduction 1347
    - Owner Tagged Value 1349
    - Properties 1277
    - Set Database Type 1348
    - Set Owner 1349
    - Set Properties 1348
    - State Machine 821
  - Table Of Contents
    - MS Word, Add 1798
  - Table Of Figures
    - MS Word, Add 1799
  - Tables in RTF Reports
    - Sections 1738
    - Split Rows 1738
  - Tag
    - Compartment, Element 659
    - Management, Advanced 767
    - Profile 1044
  - Tag Type
    - Predefined, Assign To Stereotype 1045
  - Tagged Region
    - Code Editor Find And Replace 1412
  - Tagged Value
    - Add 763
    - Add To Operations 712
    - Assign Information To 765
    - Assign To Item 764
    - Behavior Parameters 718
    - By Dragging Element From Toolbox 1029
    - Connector, Use 1046
    - Custom, Create 1113
    - Duplicate Values 762
    - Element Package, Automation Interface 1895
    - Entity Relationship Diagram 1382
    - For Oracle Table Properties 1350
    - For Schema Owner 1349
    - For Table Owner 1349
    - Fully Qualified, Show 585
    - Fully-Qualified Value 762
    - In UML Profiles 1028
    - Inherited, Show 585
    - Insert Column In Package Browser 459
    - Macros, Code Template Syntax 1129
    - Masked, Create 1113
    - Model Components And 762
    - Modify 762
    - Modify Value 764
    - Of Attributes 699
    - Quick Add 763
    - Reference Data, Create 1113
    - Show Duplicates 766
    - Structured, Create 1111
    - Synchronize, And Constraints 1029
    - Synchronize, From MDG Toolbox Pages 1029
    - Toolbar 762
    - Types 775
    - Types Of Value Field 764
    - View 762
    - What Is A? 762
    - Window 762

- Tagged Value Broadcast Events
  - Add-In Model 2031
  - EA\_OnAttributeTagEdit 2031
  - EA\_OnConnectorTagEdit 2032
  - EA\_OnElementTagEdit 2033
  - EA\_OnOperationTagEdit 2034
- Tagged Value Type
  - Add To MDG Technology 1069
  - Filters 1108
  - Introduction 1108
  - Predefined Reference Data 1111
  - Predefined Structured 1108
- Tagged Values window 1371
- TaggedValue
  - Automation Interface, Element Package 1895
- Target
  - Role 760
  - Set For Message 754
  - Types, Implementation Report 1785
- Task
  - Automation Interface Repository 1872
  - Completion 349
  - Details 357
  - Element 1299
- Task Management
  - In Project Calendar 398
- TCF
  - Value 401
  - Weighting 401
- TcSE
  - MDG Integration For, Enterprise Architect 1036
- Team
  - Develop Projects In, Introduction 185
  - Development, Introduction 135
  - Development, Project Sharing 185
- Team Deployment
  - And Version Control 247
  - Version Control Branching 247
- Team Foundation Server
  - Version Control Option 274
- Team Review 221
  - Access 216
  - Access From Shortcut 140, 142
  - Access From Shortcut (Direct Definition) 141
  - Add External File Link To Post 227
  - Add New Category 220
  - Add New Post 222
  - Add New Topic 221
  - Add Object Link To Post 227
  - Add Team Review Link To Post 227
  - Capture Diagram Image As Resource 228
  - Connections To Other Team Reviews 230
  - Context Menu Options 218
  - Copy Path To Clipboad 218
  - Create Category 220
  - Create Post 222
  - Create Reply 224
  - Create Topic 221
  - Delete Category 218
  - Delete Item 226
  - Delete Post 218
  - Delete Resource 218
  - Delete Resources 228
  - Delete Topic 218
  - Edit Item 226
  - Editor 226
  - Export Package As Resource 228
  - Hyperlink To 1292
  - Icons 216
  - Import Resource Package 228
  - Introduction 216
  - Load Data When Required 229
  - Loading Behavior 229
  - Mark All Posts Unread 229
  - Options 229
  - Postings in Model Views 464
  - Preload 229
  - Reply To Post 224
  - Resources 228
  - Save Profiler Report As Resource 1662
  - Search 229
  - Tab 216
  - View Diagram Image Resource 228
  - Was Discussion Forum 216
  - Window 216
- Teamcenter Systems Engineering, Siemens PLM
  - MDG Integration For, Enterprise Architect 1036
- Technical Complexity Factor
  - Estimate Project Size 404
  - Value 401
  - Weighting 401
- Technology Developer
  - And Enterprise Architect 65
  - Project Role 65
- Technology Event
  - EA\_OnInitializeTechnologies 2035
- Technology Events
  - Add-In Model 2034
  - EA\_OnDeleteTechnology 2038
  - EA\_OnImportTechnology 2039
  - EA\_OnPostActivateTechnology 2037
  - EA\_OnPreActivateTechnology 2036
  - EA\_OnPreDeleteTechnology 2037
- Technology File

- Technology File
  - Generate (Option) 90
  - Import (Option) 90
- Technology-Defined Model View
  - Set Up 469
- Template
  - Behavioral Model 1137
  - Editor In SDK 1147
  - Editor, Code Templates 1485
  - Element 1026
  - Fragments, HTML 1806
  - Instantiated 943
  - Model, Incorporate In Technology 1087
  - Package, Settings 643
  - Parameterized Classes 943
  - RTF, Import 1765
  - Script 1818
  - Transformation, Default 1332
- Template Binding
  - Connector 1010
- Term
  - Automation Interface Repository 1873
- Terminate
  - Element 933
- Test
  - Application, Analyzer menu Option 94
  - Automation Interface, Element Package 1896
  - Copy Between Categories 1703
  - Create Defect From 1707
  - Documentation 1708
  - Generate From Scenario 685
  - Import From Other Element 1705
  - Model Template 527
  - Move Between Categories 1703
  - Report 1708
  - Result Output 1708
  - Script Output 1708
  - Submenu (Analyzer) 97
  - Unit, In Execution Analysis 1669
  - Unit, Record Results In Execution Analysis 1670
  - Unit, Run In Execution Analysis 1670
  - Unit, Set Up In Execution Analysis 1669
- Test Case
  - Element 1674
- Test Cases
  - Generate From Scenario 685
- Test Cut 1672, 1678
  - Element 1675
- Test Details Dialog 1696
- Test Domain 1672
  - Elements And Connectors 567
  - Group, Toolbox 567
- Test Script
  - Introduction 1451
  - JUnit 1669
  - NUnit 1669
- Test Scripts
  - Compartment 1707
- Test Set 1672
  - Element 1675
- Test Suite 1672
  - Element 1676
- Tester
  - And Enterprise Architect 64
  - Project Role 64
- Testing
  - Acceptance 1700
  - Asterisk On Testing Window Tabs 1695
  - Autonaming 1695
  - Compartment, Element 659
  - Import Element Scenarios 1703
  - Import Internal Constraint 1706
  - Import Internal Requirement 1706
  - Import Package Scenarios 1703
  - Import Responsibility 1706
  - Integration 1698
  - MDG Technology For, Enterprise Architect 1036
  - Model Template 527
  - Overview 1694
  - Report, Generate 1786
  - Scenario 1701
  - Support 1694
  - System 1699
  - Type, Define 788
  - Unit 1696
  - Window 1695
  - Window, Acceptance Test Tab 1700
  - Window, Integration Test Tab 1698
  - Window, Scenario Test Tab 1701
  - Window, System Test Tab 1699
  - Window, Unit Test Tab 1696
  - Workspace 1695
- Testing Details Report
  - Generate 1786
- Testing Feature
  - Insert In Element 728
- Testpoint 1453
  - Diagram 567
  - Introduction 1672
  - Management 1672
  - Toolbox Group 567
  - Window 1676

- Testpoint 1453
  - Window Toolbar 1678
- Testpoint Management
  - Class Invariant Dialog 1678
  - Operation Constraints Dialog 1679
- Testpoints 1453
- Text Element
  - Create 639
  - Insert New From Toolbar 110
- Text File
  - Create Element From 1156
- TFS
  - Package Version Control History, In 293
  - Version Control Option 274
  - Working Folder For Enterprise Architect Version Control 274
  - Workspace For Enterprise Architect Version Control 274
- The Open Group Architecture Framework
  - MDG Technology For, Enterprise Architect 1036
- Theme
  - Microsoft Office Styles 125
  - Microsoft Visual Basic Styles 125
  - Options 125
  - Visual Style 78
- Tidy Line Angles 745
- Time Event 915
- Time Interval
  - Compress 845, 847
  - Context Menu 843
  - Copy and Paste 847
  - Create 843
  - Delete 843
  - Description 843
  - Move 843
  - Operations 847
  - Resize 843
  - Select 846
  - Shift Left Or Right 847
  - Transitions 847
- Time Range
  - Set For Timing Diagram 832
- Timeline Element States
  - Add Via Configure Timeline Dialog 840
  - Delete via Configure Timeline Dialog 840
  - Edit Via Configure Timeline Dialog 840
  - Maintain 840
  - Numeric Range Generator 841
- Timeline Range
  - Set For Timing Diagram 832
- Timeline Start Position
  - Set For State Lifeline Element 833
- Timing
  - Constraint 993
  - Details, Change 993
  - Elements and Connectors 557
  - Group, Toolbox 557
  - Message 999
  - Message, Create 1000
  - Observation 993
- Timing Diagram
  - Add Value Lifeline Element 838
  - Create 832
  - Description 830
  - Edit Options 833
  - Edit Value Lifeline Element 838
  - Elements And Connectors 830
  - Example 831
  - Set Time Range 832
- TOGAF
  - MDG Technology For, Enterprise Architect 1036
- Tomcat
  - Server, Configuration 1428
  - Server, Debugging 1425
  - Service Configuration 1428
- Toolbar
  - (File) Search, Debugging 1454
  - Add Commands 118
  - Add Menu Commands 119
  - Add Toolbar Commands 119
  - Change Command Icon Appearance 118
  - Code Generation 109
  - Console Tab, Scripting Window 1820
  - Create 119
  - Current Connector 114
  - Current Element 112
  - Customize 119
  - Debug & Record 116
  - Default Tools 107
  - Diagram 111
  - Diagram List 459
  - Display Labels 119
  - Docked 106
  - Element 110
  - Floating Buttons, Display 432
  - Floating, Structural Specification 676
  - Format (Element Appearance) 541
  - Hide 119
  - Large Icons 125
  - Manage Searches 482
  - Missing Icon 118
  - Model Search 475

- Toolbar
  - Notes 770
  - Options, Customize Appearance 125
  - Package Browser 459
  - Project 108
  - Project Browser 452
  - Remove 119
  - Remove Commands 118
  - Rename 119
  - Screen Tips 125
  - Scripting Window 1818
  - Show 119
  - Submenu 78
  - Testpoints Window 1678
  - UML Elements 110
  - Workspace 106
  - Workspace Layouts 115
- Toolbar Customize Button
  - Show/Hide 78
- Toolbox
  - Activity Group 558
  - Add Stereotype To Diagram Element 546
  - Add To MDG Technology 1068
  - Analysis Group 561
  - Appearance Options 549
  - ArchiMate Group 1272
  - BPMN 1.0 1220
  - BPMN 1.1 1220
  - BPMN 2.0 1222
  - BPMN 2.0 Business Process 1222
  - BPMN 2.0 Choreography 1224
  - BPMN 2.0 Conversation 1225
  - BPMN 2.0 Type 1227
  - BPMN Group 1219
  - Business Modeling Group 1191
  - Class Group 554
  - Collapse Page 546
  - Common Group 552
  - Communication Group 556
  - Component Group 559
  - Composite Group 555
  - Connectors For Extending In Profile 1079
  - Create Elements And Connectors 546
  - Custom Group 562
  - Customize 1074
  - Data Flow Diagram Group 1184
  - Data Modeling Group 566
  - Default, Override In Profile 1076
  - Deployment Group 559
  - Documentation Group 1774
  - Elements For Extending 1077
  - Entity Relationship Diagram 1382
  - Eriksson-Penker Group 1273
  - Expand Page 546
  - Gang Of Four Pattern Group 1545
  - GoF Pattern Group 1545
  - Hide 546
  - Hide Labels 549
  - ICONIX Group 1542
  - Interaction Group 556
  - Learning Center, Define 1083
  - Maintenance Group 564
  - MDG Technology Groups 1031
  - Metamodel Group 561
  - Mind Mapping Group 1182
  - Object Group 555
  - Override Default In Toolbox Profile 1076
  - Page Attributes 1075
  - Pin Pages 549
  - Profile Group 560
  - Profile, Create For MDG Technology 1074
  - Profiles 1074
  - Requirement Group 563
  - Set Toolbox Visibility 549
  - Shortcut Menu 551
  - Show 546
  - Show Labels 549
  - SoaML Pages 1624
  - SOMF Pages 1628
  - SPEM 407
  - State (Machine) Group 558
  - Synchronize Stereotyped Tagged Values 546
  - SysML Groups 1552
  - SystemC Group 1474
  - Test Domain Group 567
  - Timing Group 557
  - Unpin Pages 549
  - Use Case Group 553
  - User Interface Group 564
  - Verilog Group 1477
  - VHDL Group 1478
  - WSDL Group 565
  - XML Schema Group 566
- Toolbox Profile
  - Connectors For Extending 1079
  - Create Hidden Submenu In 1075
  - Items, Assign Icons For 1076
  - Pages That Can be Overridden 1077
- Tools
  - Custom 121
  - External Applications 121
  - Menu 90
  - Pass Parameters To External Tools 123
- Top Margin

- Top Margin
  - Sequence Diagram, Change 858
- Topic
  - Add To Team Review 221
  - Create 221
  - Delete 218
- TortoiseSVN
  - In Version Control 260
- Trace
  - Connector 1012
  - Relationship 1012
  - Statements 1440
- Trace Statements
  - Output 1435
- Trace Tools 440
- Traceability 494
  - Diagrams 505
  - Element Grouping 506
  - For Diagram 495
  - For Elements 495
  - In Requirements Models 1174
  - Introduction 493
  - Model Structure 506
  - Package Organization 506
  - Themes 493
  - Tools 494
  - Window 495
  - Window, In Traceability 494
  - With Dependency Report 494
  - With Implementation Report 494
  - With Relationship Matrix 494
  - With Traceability Window 494
- Tracepoint
  - Statement Output 1435
- Tracepoints
  - In Code 1440
  - Set 1440
- Track Changes
  - Auditing 298
  - Baselines 298
  - Introduction 298
- Trademarks 23
- Transfer
  - Project Data Between Repositories 343
- Transform
  - Connector End 1340
  - Connectors 1338
  - Copy Information 1340
  - Duplication Of Connectors 1338
  - Elements, MDA-Style Transformations 1305
  - Model, MDA-Style Transformations 1305
  - Names 1341
  - Objects 1333
- TRANSFORM\_CLASSIFIER
  - Macro 1342
- TRANSFORM\_CURRENT
  - Macro 1340
- TRANSFORM\_REFERENCE
  - Macro 1338, 1342
- Transformation
  - Built In, MDA-Style Transformation 1307
  - C# 1307
  - Data Model To ERD 1310
  - DDL 1310
  - Dependencies 1301
  - EJB 1314
  - Entity Bean 1314
  - ERD To Data Model 1317
  - Java 1319
  - JUnit 1321
  - NUnit 1323
  - Session Bean 1314
  - Write 1332
  - WSDL 1325
  - XSD 1326
- Transformation Dependency
  - Trace With Traceability Window 495
- Transformation Template
  - Default 1332
  - Modify 1330
  - Transfer Between Models 1331
- Transition
  - Add To State Lifeline Elements 835
  - Add Via Configure Timeline Dialog 842
  - Change In State Machine Table 825
  - Change Time, State Lifeline Element 835
  - Connector 1013
  - Delete On State Lifeline Element 835
  - Delete Via Configure Timeline Dialog 842
  - Edit In Time Intervals 847
  - Edit On State Lifeline Elements 835
  - Edit Via Configure Timeline Dialog 842
  - Effect 1013
  - Guard 1013
  - Highlight Associated Trigger or State 825
  - Insert In State Machine Table 825
  - Locate In State Machine Diagram 827
  - Locate In State Machine Table 827
  - Merge On State Lifeline Element 835
  - Move On State Lifeline Elements 835
  - Properties 1013
  - Relationship 1013
  - State Machine Table Conventions 828
  - Trigger 1013

- Transitions Collection
    - Automation Interface, ElementFeatures Package 1910
  - Translation
    - RTF Report Generation 1773
  - Tree Style Hierarchy
    - Create 755
    - Set Default Link Style 755
  - Trial Version
    - Extend Trial Period 13
    - Of Enterprise Architect 13
    - Select Edition To Trial 13
    - Select Workspace Layout 13
  - Triangle
    - Red 412
  - Tricks and Traps
    - Create Add-In 1972
  - Trigger
    - Create 934
    - Create In State Machine Table 825
    - Create In Transition Properties 1013
    - Element 934
    - Find In Project Browser 1013
    - For Transition 1013
    - Intermediate Event, BPEL 1242
    - Locate In State Machine Diagram 827
    - Locate In State Machine Table 827
    - Operation, What Is A? 1366
    - Ports 934
    - Properties Tab 934
    - Reposition In State Machine Table 826
    - Start Event, BPEL 1240
    - State Machine Table Conventions 828
    - Type 1013
  - Trigger Operation
    - Create 1366
    - What Is A? 1366
  - Trigger Property
    - Find In Project Browser 737
  - Type Hierarchy Dialog 628
  - Type Libraries
    - For Script Editor 1407
  - Type Specific Menu Section
    - Connector 737
- U -
- UAC
    - And Debugging 1418
  - UI Control
    - Element 1282
  - Ultimate Edition
    - Of Enterprise Architect 14
  - UML
    - 1.3 319, 320, 414
    - 1.3, Import From XMI 323
    - 1.4 320
    - 1.4, Import From XMI 323
    - 1.5 432
    - 2.0 Migration 414
    - 2.0, Import From XMI 323
    - 2.3 - Definition 7
    - Analysis Tool - Enterprise Architect 5
    - Build Systems 7
    - Connectors 966
    - Data Modeling Profile 1276
    - Definition 793
    - Design Systems 7
    - Design Tool - Enterprise Architect 5
    - Dictionary 793
    - DTD 326
    - Elements 864
    - Enterprise Architect Modeling Platform 515
    - Extend 793
    - Manage Complexity 7
    - Mappings To XSD 1597
    - Model Complexity 7
    - Model Structure, Manage - Enterprise Architect 7
    - Models Under Single Root 324
    - Pattern 1021
    - Recommended Reading 793
    - Shared Model Development - Enterprise Architect 7
    - Support - Enterprise Architect 7
    - Syntax Checking Option 426
    - Visualize Systems - Enterprise Architect 7
  - UML Behavioral Diagram
    - Overview 811
  - UML Business Process Model
    - In Enterprise Architect 520
  - UML Class Model
    - In Enterprise Architect 523
  - UML Component Model
    - In Enterprise Architect 525
  - UML Database Model
    - In Enterprise Architect 524
  - UML Deployment Model
    - In Enterprise Architect 526
  - UML Diagram
    - Add To Project 568
    - Copy, Deep 583
    - Copy, Shallow 583
    - Create 568

- UML Diagram
  - Duplicate 583
  - Extended 795, 1181
  - In Enterprise Architect 538
  - MDG Technology 795
  - Overview 795
  - Paste 583
  - Types 795
  - What Is A? 795
- UML Domain Model
  - In Enterprise Architect 522
- UML Element
  - Behavioral Diagram Elements 864
  - Structural Diagram Elements 940
  - Toolbar 110
- UML Maintenance Model
  - In Enterprise Architect 528
- UML Model
  - Add Connectors In Enterprise Architect, Tutorial 42
  - Add Diagram In Enterprise Architect, Tutorial 40
  - Add Element In Enterprise Architect, Tutorial 41
  - Add Element To Diagram, Tutorial 41
  - Add Package In Enterprise Architect, Tutorial 39
  - Add View, Tutorial 38
  - CSV Import And Export In Enterprise Architect 337
  - Define Connector Properties In Enterprise Architect 43
  - Define Element Properties In Enterprise Architect 43
  - Project Roles And Tasks In Enterprise Architect 57
  - Specialized 1178
- UML Model Pattern
  - Introduction 519
- UML Model Template
  - In Enterprise Architect 519
  - Introduction 519
- UML Modeling
  - And MDG Technologies 515
  - And Requirements Management In Enterprise Architect 515
  - Build Models 515
  - Business Modeling In Enterprise Architect 515
  - What Is? 515
  - With Enterprise Architect 515
  - With Packages 533
  - With UML Patterns In Enterprise Architect 515
  - With UML Profiles In Enterprise Architect 515
  - With UML Stereotypes In Enterprise Architect 515
- UML Modeling Tool
  - Enterprise Architect - Key Features 10
- UML Pattern
  - Actions 1022
  - Add To Diagram 1024, 1025
  - Create From Diagram 1022
  - In Resources View 1022
  - Save 1022
  - Save From Diagram 1022
- UML Profile
  - And Element Templates 1026, 1039
  - Import 1027
  - Import From XML 1026, 1039
  - Stereotypes 1026, 1039
  - Synchronize Elements 1029
  - Synchronize Stereotypes 1029
  - Synchronize Tagged Values And Constraints 1029
  - XSD, Stereotypes 1602
- UML Project Model
  - In Enterprise Architect 528
- UML Requirements Model
  - In Enterprise Architect 521
- UML Resources
  - Patterns 1021, 1023
- UML Structural Diagram
  - Overview 796
- UML Syntax Compliance
  - Turn Off 1689
- UML Testing Model
  - In Enterprise Architect 527
- UML Types
  - Cardinality (Multiplicity) 776
  - Dialog 772
  - Stereotypes 773
  - Tagged Values 775
- UML Use Case Model
  - In Enterprise Architect 521
- UML\_EA.DTD 326
  - File 323
- UML2
  - File Import 324
- Unadjusted Use Case Points 404
- Undo
  - Last Action, Diagram Edits 603
  - Option (Edit Menu) 74
- Unicode Support 35, 422
- Unified Modeling Language 793
- Unique
  - Constraint 1366

- Unique
  - Index 1366
- Unit Test Command
  - Introduction 1451
- Unit Test Script
  - Create 1451
- Unit Testing 1696
  - Create Test Scripts, Execution Analysis 1669
  - Define Tests, Execution Analysis 1669
  - Introduction, Execution Analysis 1669
  - JUnit 1669
  - NUnit 1669
  - Record Test Results, Execution Analysis 1670
  - Run, Execution Analysis 1670
  - Set Up, Execution Analysis 1669
- Unlock
  - Connector 207
  - Element 207
- Unpin
  - Toolbox Pages 549
- Update
  - Element Phase, For Package 411
  - Element Status, For Package 411
  - Element Version, For Package 411
  - Package Phase 411
  - Package Status 411
  - Package Version 411
- Upgrade
  - Existing License (v. 6.5 And Earlier) 2099
  - Existing License (v. 7.0+) 2099
  - Model 417
  - Model, Upgrade Wizard 417
  - Project 417
  - Replicas 193, 417
  - Wizard 417
- Upsize
  - From Desktop and Professional Editions 148
  - To Access 2007 174
  - To MySQL 172
  - To Oracle 177
  - To PostgreSQL 179
  - To Progress OpenEdge 182
  - To SQL Server 176
  - To SQL Server Desktop Engine (MSDE) 182
  - To Sybase Adaptive Server Anywhere (ASA) 180
- Usage
  - Connector 1014
  - Of Element 630
  - Relationship 1014
- Use
  - Classifiers 706
  - Connector 1014
  - Pattern 1024
  - Relationship 1014
  - Spell Checker 376
- Use Case
  - And Requirements 1164
  - Arrowhead, Show 754
  - Create For Method 1406
  - Element 935
  - Elements and Connectors 553
  - Extension Points 937
  - Group, Toolbox 553
  - Keyword 404
  - Link To Method 1406
  - Metrics 400
  - Metrics Dialog 404
  - Model Template 521
  - Phase 404
  - Points, Unadjusted 404
  - Scenarios 853
  - View 530
- Use Case Diagram
  - Description 813
  - Elements And Connectors 813
  - Example 815
- Use Element Extras
  - Automation Interface Code Example 1958
- Use Repository Extras
  - Automation Interface Code Example 1961
- User
  - Default Fonts 429
  - Defined Searches, Storage 484
  - Dictionary, For Spell Checker 377
  - Directory 778
  - Forum 31
  - Groups 198
  - ID, Import From Active Directory 199
  - Settings 1524
- User Interface
  - Components 69
  - Control Element 1282
  - Element 1282
  - Elements and Connectors 564
  - Group, Toolbox 564
  - MDG Win32® User Interface Technology 1286
  - Screen Prototype 1281
- User Interface Diagram
  - Description 1281
  - Elements And Connectors 1281
  - Example 1286
- User Lock
  - Identify Owner 215

User Lock  
 Indicators 214  
 User Security  
 Add Connector To Locked Element 211  
 Apply User Lock 213  
 Assign User To Group 201  
 Basics 194  
 Change Password 209  
 Disable 196  
 Enable 196  
 Identify Lock Owner 215  
 Import User ID From Active Directory 199  
 List Of Permissions 205  
 Lock Elements & Diagrams 210  
 Lock Packages 212  
 Locked Element Indicators 214  
 Maintain Groups 203  
 Maintain Users 198  
 Manage User-level Locks. 215  
 Password Encryption 208  
 Policy 197  
 Re-enable 196  
 Release Elements & Diagrams 210  
 Reset Password 209  
 Set Group Permissions 204  
 Set Password 209  
 Set Up Single Permissions 201  
 Tasks 194  
 View All Permissions 202  
 View And Manage Locks 207  
 What Is User Security? 194  
 User Security Groups  
 Assign User To 201  
 Set Up 201  
 User Settings  
 Options Dialog 118  
 User/Group Lock  
 Release 210  
 Set 210  
 User/Group Lock Policy 197  
 Uses Context List Facility 674  
 Using Enterprise Architect  
 Application Workspace 69  
 Remove Recent Project 71  
 Start Page 70  
 Utility  
 Compare 308  
 Diff 308  
 UUCP  
 Estimate Project Size 404

## - V -

Validate  
 Business Rules 1216  
 Package Configuration For Version Control 296  
 Rule Composer 1216  
 Validation  
 Cancel 81  
 Of Diagram 1688  
 Of Element 1688  
 Of Model, Configure 1689  
 Of Model, Configure For MDG Technology 1086  
 Of Package 1688  
 Rules, Model 1690  
 Validation, Properties  
 Attribute 1692  
 Element 1692  
 Feature 1692  
 Relationship 1692  
 Validation, Well Formedness  
 Attribute 1691  
 Diagram 1691  
 Element 1691  
 Feature 1691  
 Relationship 1691  
 Value Lifeline Element 938  
 Add States 838  
 Add To Timing Diagram 838  
 Add Transitions 838  
 Change Transition Time 838  
 Delete Transitions 838  
 Edit Transitions 838  
 Sizing and Scale 838  
 States 838  
 Transitions 838  
 ValueOf Field  
 RTF Style Template Editor 1742  
 Variable  
 Comparison Of Values 1444  
 Debug, Break On Change In Value 1445  
 Definitions, Code Template Syntax 1145  
 Definitions, Examples 1145  
 References, Code Template Syntax 1145  
 References, Examples 1145  
 Snapshots 1444  
 VB  
 Set Up In Automation Interface 1825  
 VB.NET  
 Code Generation 1537  
 Language Options 1537

- VB.NET
  - Modeling Conventions 1475
- VBScript 1817, 1818
- Verilog
  - Code Generation 1538
  - Enterprise Architect Toolbox Pages 1477
  - Language Options 1538
  - Modeling Conventions 1477
- Version Control
  - And Reference Data 246
  - And Replication 275
  - And Sequence Diagram 854
  - Apply To Enterprise Architect Model 245
  - Apply To Model Branch 289
  - Basics 246
  - Branching 247
  - Configuration 253, 268, 270, 291
  - Configuration, Team Deployment 247
  - Configuration, Use Previously-Defined 269
  - Configure Current Package 277
  - Configure Package 282
  - Considerations 243
  - Controlled Packages 327
  - Copy-Modify-Merge Policy 246
  - CVS Options 272
  - Discussion Of File Control 245
  - Export Model Branch 290
  - Features 242
  - File History 278
  - File Properties 278
  - Import Model Branch 291
  - In Team Deployment 247
  - Include Other Users' Packages 289
  - Introduction 243
  - Locking - Necessary? 246
  - Lock-Modify-Unlock Policy 246
  - Manual, With XML 334
  - Menu (Package) 278
  - Menu (Project) 277
  - Nested Packages 245
  - Offline 250
  - Package Configuration, Validate 296
  - Policies 246
  - Project Browser Indicators 249
  - Recommendations 275
  - Refresh View Of Shared Model 187
  - Resynchronize Package Version Control Status 297
  - Resynchronize Status Of All Packages 277
  - Review Version Control History (SCC) 294
  - Review Version Control History (TFS, CVS, Subversion) 293
  - Revision, Check Out 293
  - SCC Options 271
  - SCC, Providers Dialog SCC, Providers Dialog 271
  - Set Up 268
  - Settings 270, 277
  - Setup Menu 277
  - Submenu (Project Menu) 82
  - Subversion Options 273
  - Subversion, Create Local Working Copy 256
  - Subversion, Create New Repository Subtree 256
  - Subversion, TortoiseSVN 260
  - Subversion, Using With Enterprise Architect Under WINE Crossover 258
  - System Requirements 253
  - Tasks 275
  - TFS Option 274
  - Usage Scenarios 244
  - Use Nested Version Control Packages 248
  - Using 275
  - Validate Package Configurations 277
  - Who Has Checked Out A Package? 278
  - Work Offline 277
- Version Controlled Package
  - Icon 454
- VHDL
  - Code Generation 1538
  - Enterprise Architect Toolbox Pages 1478
  - Language Options 1538
  - Modeling Conventions 1478
- View
  - Add To UML Model, Quick Start 38
  - Audit 302
  - Create, Data Modeling 1370
  - Database 1370
  - Diagram List 462
  - Element, Data Modeling 1370
  - Locks 207
  - Menu 76
  - Next Diagram 600
  - Of Model 464
  - Package Browser 456
  - Previous Diagram 600
  - Project Statistics 80
  - Report On, Data Modeling 1370
  - Requirements 1170
  - Submenus 78
  - Tabs, Close 543
- View Elements Of Array
  - Debugger 1443
- View Header

- View Header
  - Model Search 480
- View Options
  - Diagram List 462
  - Diagram View 540
  - Package Browser 456
- View Tab Switcher 545
- Views
  - Add 531
  - Class 530
  - Component 530
  - Delete 532
  - Deployment 530
  - Dynamic 530
  - Manage 530, 531
  - Rename 531
  - Simple 530
  - Use Case 530
- Virtual Document
  - Add Packages As Attributes 1778
  - Change Package Sequence 1780
  - Consecutive Numbering Through 1782
  - Create Master Document For 1776
  - Create Model Document For 1777
  - Delete Package Attributes 1779
  - Delete Packages 1779
  - Document Order 1780
  - Generate HTML Report From 1814
  - Generate RTF Report From 1782
  - Introduction 1774
  - Master Document Element 1774
  - Model Document Element 1774
  - Model Search Sequence 1780
  - Move Package Attributes Between Elements 1780
  - Numbering Through 1782
  - RTF Bookmarking In 1774
  - Sequence of Model Document Elements 1780
  - Sequence of Package Attributes 1780
  - Tagged Values For Master Document 1776
- Virtual Document (HTML) 1813
- Virtual Machine 1422
- Visibility Indicators 574
  - Values 1457
- Visibility Of Elements
  - Customize 638
- Visible Class Members
  - Set Diagram Appearance Options 575
- Visio
  - MDG Link For, Enterprise Architect 1036
- Vista
  - Permissions For Enterprise Architect 27
- Visual Basic
  - Code Generation 1539
  - Connect To Automation Interface 1823
  - Debugging, General Setup 1421
  - Import, Reverse Engineering 1510
  - Language Options 1539
  - Modeling Conventions 1480
  - Set Up In Automation Interface 1825
- Visual Basic.Net
  - Import, Reverse Engineering 1510
- Visual Execution Analyzer
  - Access 1635
  - Add State Transitions 1653
  - Advanced Debug Techniques, Java 1423
  - Availability 1635
  - Break On Variable Changing Value 1445
  - Breakpoint Management 1436
  - Breakpoint Storage 1441
  - Breakpoints And Markers Window, Record Sequence Diagrams 1646
  - Build Script, Create 1415
  - Control Recording, Record Sequence Diagrams 1648
  - Create Sequence Diagram, Call Stack 1450
  - Debug .NET 1428
  - Debug .NET CLR Versions 1429
  - Debug .NET With COM Interop Process 1430
  - Debug Another Process 1448
  - Debug Apache Tomcat Server Configuration 1428
  - Debug Apache Tomcat Windows Service 1428
  - Debug ASP .NET 1431
  - Debug Java 1422
  - Debug Java Applets In Internet Browsers 1424
  - Debug Java Web Servers 1425
  - Debug JBOSS Server Configuration 1427
  - Debug Symbols, C++ And Native Applications 1421
  - Debugger Frameworks 1416
  - Debugger System Requirements 1417
  - Debugger, Overview 1416
  - Debugging Actions 1441
  - Deploy Script, Create New 1453
  - Deploy Script, Introduction 1453
  - Diagram Features, Generate Sequence Diagrams 1652
  - Difference Between Recording Marker And Breakpoint 1642
  - File Search, Introduction 1454
  - File Search, Use 1454
  - For C++ Applications 1420
  - For Microsoft Native Applications 1420

- Visual Execution Analyzer
  - For WINE Applications 1419
  - General Debug Setup, Java 1422
  - Generate Sequence Diagram 1651, 1652
  - Inspect Process Memory 1445
  - Introduction 1635
  - Java Debug Session, Attach To VM 1424
  - Map State Changes 1657
  - Marker Management 1436
  - Marker Storage 1441
  - Marker Types, Record Sequence Diagrams 1642
  - MDDE Basic Setup 1392
  - MDDE External Tools 1392
  - MDDE, Analyzer Scripts, Manage 1392
  - MDDE, Build Scripts 1415
  - MDDE, Generate Code 1392
  - MDDE, Script Actions, Define 1394
  - MDDE, Synchronize Code 1392
  - Model Simulation 1681
  - Object Workbench, Create Variables 1665
  - Object Workbench, Introduction 1663
  - Object Workbench, Invoke Methods 1666
  - Object Workbench, Overview 1664
  - Object Workbench, Variables 1665
  - Outputs 1635
  - Overview 1635
  - Place Markers, Recording Sequence Diagrams 1642
  - Profiler Overview 1658
  - Record Activity For Class 1646
  - Record Activity For Method 1645
  - Record Sequence Diagrams, Introduction 1637
  - Record Sequence Diagrams, Overview 1638
  - Record Sequence Diagrams, Prerequisites 1639
  - Record Sequence Diagrams, Recording Options 1639
  - Record Sequence Diagrams, Set Up 1639
  - Record State Changes 1657
  - Record Unit Test Results 1670
  - Recording Actions 1449
  - Recording History 1651
  - Recording Markers, Activate, Record Sequence Diagrams 1647
  - Recording Markers, Disable, Record Sequence Diagrams 1647
  - Recursive Builds 1416
  - Run Script, Create New 1451
  - Run Script, Introduction 1451
  - Run Unit Test 1670
  - Save Call Stack 1450
  - Save Recording History 1653
  - Script Search 1454
  - Search Window 1454
  - Sequence Diagrams, Limit Auto Recording 1642
  - Set Code Breakpoint 1438
  - Set Data Breakpoint 1438
  - Set Up Debug Session 1417
  - Set Up Debug Session For .NET 1429
  - Set Up To Capture State Changes 1653
  - Show Loaded Modules 1446
  - Show Output 1446
  - Start Debugger 1447
  - State Machine Diagram 1655
  - State Transition Diagram 1655
  - Step Into Function Calls 1449
  - Step Out Of Functions 1449
  - Step Over Lines Of Code 1448
  - Step Though Function Calls 1449
  - Stop Debugger 1447
  - Tooltips In Code Editor 1447
  - UAC-Enabled Operating Systems 1418
  - Unit Test Script, Create 1451, 1669
  - Unit Test Script, Introduction 1451
  - Unit Test, Record Results 1670
  - Unit Testing, Introduction 1669
  - Uses Of 1635
  - View Call Stack 1441
  - View Local Variables 1442
  - View Local Variables, Long Values 1443
  - View Variables In Other Scopes 1444
  - Work With Marker Sets, Record Sequence Diagrams 1648
  - Workspace Layouts 1391
- Visual Execution Profiler
  - Attach To Process 1660
  - Getting Started 1659
  - Launch 1660
  - Operation 1661
  - Overview 1658
  - Prerequisites 1658
  - Report, Example 1658
  - Report, Load 1662
  - Report, Save 1662
  - Report, Save As Resource In Team Review 1662
  - Set Options 1661
  - Set Sample Intervals 1661
  - Start 1660
  - Stop 1660
  - Supported Platforms 1658
  - System Requirements 1658

- Visual Execution Profiler
    - Team Review, Save Report As Resource 1662
    - Toolbar 1659
  - Visual Execution Sampler
    - Attach To Process 1660
    - Getting Started 1659
    - Launch 1660
    - Operation 1661
    - Overview 1658
    - Prerequisites 1658
    - Report, Example 1658
    - Report, Load 1662
    - Report, Save 1662
    - Report, Save As Resource In Team Review 1662
    - Set Options 1661
    - Set Sample Intervals 1661
    - Start 1660
    - Stop 1660
    - Supported Platforms 1658
    - System Requirements 1658
    - Team Review, Save Report As Resource 1662
    - Toolbar 1659
  - Visual Representation
    - Of Elements, Introduction 658
  - Visual Studio 2005/2008
    - MDG Integration For, Enterprise Architect 1036
  - Visual Studio.NET
    - MDG Link For, Enterprise Architect 1036
  - Visual Styles
    - Microsoft Office Styles 125
    - Microsoft Visual Basic Styles 125
    - Options 125
    - Select 125
    - Submenu 78
    - Themes 78
    - View Menu Option 76
  - Visualize Systems
    - Enterprise Architect 7
  - VM
    - Attach To In Java Debug Session 1424
- W -
- W3C XSD
    - Elements and Connectors, Enterprise Architect 566
  - W3CWSDL
    - Elements And Connectors, Enterprise Architect 565
  - WAN Optimization
    - Enterprise Architect Performance 162, 163, 164, 166, 167, 169, 170
    - For ASA Data Repository 169
    - For MSDE Server Data Repository 170
    - For MySQL Data Repository 162
    - For Oracle Data Repository 164, 166
    - For PostgreSQL Data Repository 167
    - For Progress OpenEdge Data Repository 170
    - For SQL Server Data Repository 163
  - WAN Optimizer
    - Introduction 183
    - Performance Of Enterprise Architect 183
    - Transmission Diagram 183
  - Watched Items
    - Debugger 1444
  - Watches Window 1444
    - Break On Variable Changing Value 1445
  - Watermark
    - Diagrams 427
  - Web
    - Page Modeling 1284
    - Report Style Templates 1804
    - Stereotypes 1284
    - Templates, Reporting 1804
  - Web Browser
    - Home Website, Access 128
    - Internet Email, Access 128
    - Web Search Engine, Access 128
  - Web Page
    - Prototype 1281
  - Web Server
    - Java, Debug 1425
  - Web Service
    - Create In BPEL 1251
    - Submenu (Tools Menu) 92
  - Web Service Definition Language 1611
  - Web Services (WSDL) 1611
    - Generate WSDL 1621
    - Import WSDL Files 1622
    - Model WSDL 1612
    - Model WSDL, Binding 1617
    - Model WSDL, Document 1620
    - Model WSDL, Message 1613
    - Model WSDL, Message Part 1614
    - Model WSDL, Port Type 1614
    - Model WSDL, Port Type Operation 1615
    - Model WSDL, Service 1619
  - Web Services Business Process Execution Language (WS-BPEL) 1235
  - Web Site
    - Access Any 128
    - Access Home 128

- Web Site
  - Home, Define Default 422
  - Sparx Systems 30
- website SOAML document page. 1624
- Well Formedness Validation
  - Attribute 1691
  - Diagram 1691
  - Element 1691
  - Feature 1691
  - Relationship 1691
- What Is
  - A Check Constraint? 1366
  - A Connector? 966
  - A Foreign Key? 1361
  - A Model? 518
  - A Pattern? 1021
  - A Primary Key? 1358
  - A Project? 137
  - A Requirement? 1152
  - A Stored Procedure? 1366
  - A Tagged Value 762
  - A Trigger Operation? 1366
  - An Index? 1366
  - Enterprise Architect? 5
  - User Security? 194
  - XMI? 319
- Whitespace
  - In Package Names 1495
- Wide Area Network (WAN) Optimizer
  - Introduction 183
  - Performance Of Enterprise Architect 183
  - Transmission Diagram 183
- Win32® User Interface
  - MDG Technology 1286
- Window
  - Autohide 105
  - Breakpoints And Markers 1646
  - Bring To Top 100
  - Call Stack 1441, 1447
  - Close All 100
  - Close All Except Current 100
  - Close Current 100
  - Combine In Frame 103
  - Connections 504
  - Constraints 689
  - Diagram Filter 491
  - Dock 103
  - Element Browser 687
  - Floating 103
  - Generalized Functions 127
  - Hide 100
  - Layout Tools 604
  - Learning Center 72
  - Links 504
  - Locals 1442
  - Locals, View Long Values 1443
  - Maintenance 1713
  - Make Active 100
  - Memory Viewer 1445
  - Menu 100
  - Model Views 464
  - Modules 1446
  - Notes 769
  - Output 127
  - Output, Debugger 1446
  - Pan And Zoom 473
  - Personal Information 377
  - Project Management 347
  - Project Task Allocation 366
  - Properties (Element) 689
  - Record & Analyze 1651
  - Relationships 504
  - Reload 100
  - Remove From Frame 103
  - Requirements 689
  - Resources 789
  - Reveal Autohidden 105
  - Scenarios 689
  - Scenarios & Requirements 689
  - Scripting 1817
  - Search, Debugging 1454
  - System 356
  - Tagged Value 762
  - Team Review 216
  - Testpoints 1676
  - Traceability 495
  - Watches 1444
  - Watches, Break On Variable Changing Value 1445
  - Workbench 1665
- Windows
  - Authentication 198
  - Service, Apache Tomcat 1428
- Windows 7
  - Use Debugger 1418
- Windows Active Directory
  - Import User Login ID From 199
- Windows Authentication
  - Accept 199
- Windows Vista
  - Permissions For Enterprise Architect 27
  - Use Debugger 1418
- WINE
  - Debugging 1419

- WINE
    - DIB Data Access Violation 1419
  - WINE-Crossover
    - Using Subversion With Enterprise Architect Under 258
  - Word
    - Special Considerations 1798
  - Word Substitution
    - RTF Report Generation 1773
  - Work Flow
    - Monitor With Model Views 413
  - Work With
    - Attributes, Automation Interface Code Example 1964
    - Enterprise Architect 5
    - Methods, Automation Interface Code Example 1965
  - workbench 1663
  - Workbench Variables
    - Constraints 1665
    - Constructors 1665
    - Create 1665
    - Delete 1665
    - Requirements 1665
  - Workbench Window 1665
  - Workflow
    - Data Structures 232
    - Monitor 386
    - Objects 232
    - Script Functions 232
    - Script, Import 1818
    - Scripts 1819
    - Scripts, Execute 386
    - Scripts, Introduction 231
    - Tab, Personal Information Window 386
  - Workflow Script
    - Search Functions With User Tasks 234
    - System-Filled Data Structures 234
    - User Filled Data Structures 235
    - User Input Validate And Control Functions 233
    - User-Called Functions 236
  - Working
    - With MDG Technologies 1031
    - With UML Connectors 736
  - Working Sets
    - Create 387
    - Delete 387
    - Open 387
    - Overview 387
    - Tab, Personal Information Window 387
  - Workspace
    - Toolbars, Docked 106
    - Toolbars, Introduction 106
  - Workspace Layout
    - For Execution Analysis 1391
  - Workspace Layouts
    - Apply 115
    - Capture 115
    - Customize 115
    - Delete 115
    - Manage 115
    - Select 115
    - Show/Hide From Toolbar 115
    - System Provided 115
    - Toolbar 115
    - Upgrade Customized 115
    - User Defined 115
    - View Menu Option 76
  - WS-BPEL 1235
  - WSDL
    - Binding Diagram 1617
    - Binding Element 1617
    - Document Element 1620
    - Elements And Connectors 565
    - Group, Toolbox 565
    - Import 1621, 1622
    - Message Diagram 1613
    - Message Element 1613
    - Model 1612
    - Port Type Diagram 1614
    - Port Type Element 1614
    - Port Type Operation 1615
    - Service Diagram 1619
    - Service Element 1619
    - Split Files 1622
    - Toolbox Pages 1612
    - Transformation 1325
    - Web Services 1612
  - WSDL Support
    - Introduction 1611
- X -
- XMI
    - Export 319, 320
    - Export MOF To 1632
    - Import 319, 323
    - Import And Auditing 307
    - Import MOF From 1632
    - Limitations 325
    - Manual Version Control 334
    - Report Deletion Of Cross Package References 335
    - Specifications 319

## XMI

UML DTD 326

XMI 1.1 320

XMI 2.1 320

## XMI Import

Impact On Auto Increment Columns 416

## XMISchema Enum

Automation Interface 1838

## XML

Code Page 436

Default XML Version 436

Documents, Default Editor 436

Import Referenced Schema 1600

Package 327

Pattern File 1021, 1022

Set Default XML Directory 436

Set Default XML Editor 436

Specifications, Options Dialog 436

Structure Tree In Source Code Viewer 1408

XMI 1.0 Prefix 436

## XML Engineering 1580

## XML Schema

Elements and Connectors 566

Generate In Garden Of Eden Style 1599

Group, Toolbox 566

Submenu (Tools Menu) 92

UML Profile For XSD 1602

XSD 1581

## XSD

Abstract Models 1595

Datatype Packages 1610

Generate 1598

Import 1600

Import, Global Element Behaviour 1601

Model 1581

Transformation 1326

XML Schema 1581

## XSDany 1602

## XSDattribute 1602

## XSDattributeGroup 1602

## XSDchoice 1602

## XSDcomplexType 1602

## XSDelement 1602

## XSDgroup 1602

## XSDrestriction 1602

## XSDschema 1602

## XSDsequence 1602

## XSDsimpleType 1602

## XSDtopLevelAttribute 1602

## XSDtopLevelElement 1602

## XSDunion 1602

**- Z -**

## Zachman

Profile 592

## Zachman Framework

MDG Technology For, Enterprise Architect  
1036

## Zoom

Control On Status Bar 117

Diagram From Toolbar 111

Diagram View 599

Submenu 599

## Zooming

Code Editor, Common 1398

## Z-Order

Diagram Structure 538

Element 584

Modify, Of Elements 538

Submenu 88



# Enterprise Architect User Guide

[www.sparxsystems.com](http://www.sparxsystems.com)